



TUGAS AKHIR - KI091391

APLIKASI REMOTE KOMPUTER BERBASIS ANDROID MENGUNAKAN WIFI DENGAN *MULTI-TOUCH*

**M. SYAIFULLAH
NRP 5107100159**

**Dosen Pembimbing I
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.**

**Dosen Pembimbing II
HENNING TITI CIPTANINGTYAS, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - KI091391

COMPUTER REMOTE APPLICATIONS BASED ON ANDROID USING WIFI WITH MULTI-TOUCH

**M. SYAIFULLAH
NRP 5110100159**

**Supervisor I
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.**

**Supervisor II
HENNING TITI CIPTANINGTYAS, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015**

LEMBAR PENGESAHAN

Aplikasi Remote Komputer Berbasis Android Menggunakan Wifi dengan *Multi-touch*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

M. SYAIFULLAH

NRP. 5107100159

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
(Pembimbing 1)
2. Henning Titi Ciptaningtyas, S.Kom., M.Sc.
(Pembimbing 2)



SURABAYA
JULI, 2015

Computer Remote Applications Based On Android Using Wifi with *Multi-touch*

Student's Name : M. Syaifullah
Student's ID : 5107100159
Department : Teknik Informatika – FTIf ITS
Supervisor I : Dr.Eng. Radityo Anggoro, S.Kom.,
M.Sc.
Supervisor II : Henning Titi Ciptaningtyas, S.Kom.,
M.Kom.

Abstract

Smartphones have evolved into an important part for all walks of society. In its development Android smartphones being the most popular. Android itself is operating system that is open source. Which meant that Android OS is free, therefore allowing the programmer to develop or create Android-based applications.

Generally an Android phone is equipped with Wi-Fi technology for wireless needs. By utilizing Wi-Fi technology and touch screen features shared by all Android phones can be made an application instead of a mouse or touchpad to control the computer remotely with Wifi as connecting object.

In this final project created a remote computer applications with utilization of a touch screen in Android smartphone instead of a mouse or touchpad that comes with the function of keyboard on computer. Applications are made to run on computers for server application and mobile android for the client application.

This application can facilitate users in performing the presentation and control the computer from a distance. As well as with utilization of username, password, and the MAC address in order to prevent outsiders to control the user's computer.

Keywords: *Smartphone, Android, Multi-touch, Remote Computer, Wifi.*

Aplikasi Remote Komputer Berbasis Android Menggunakan Wifi dengan *Multi-touch*

Nama : M. Syaifullah
NRP : 5107100159
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Dr.Eng. Radityo Anggoro, S.Kom.,
M.Sc.
Dosen Pembimbing II : Henning Titi Ciptaningtyas, S.Kom.,
M.Kom.

Abstrak

Smartphone telah berkembang menjadi bagian yang penting bagi berbagai kalangan masyarakat. Dalam perkembangannya sekarang ini *smartphone* Android menjadi yang paling diminati. Android sendiri adalah *operating system* yang bersifat *open source*. Yang dimaksud *open source* bahwa OS Android adalah gratis, sehingga memungkinkan para programmer untuk mengembangkan atau membuat aplikasi berbasis Android.

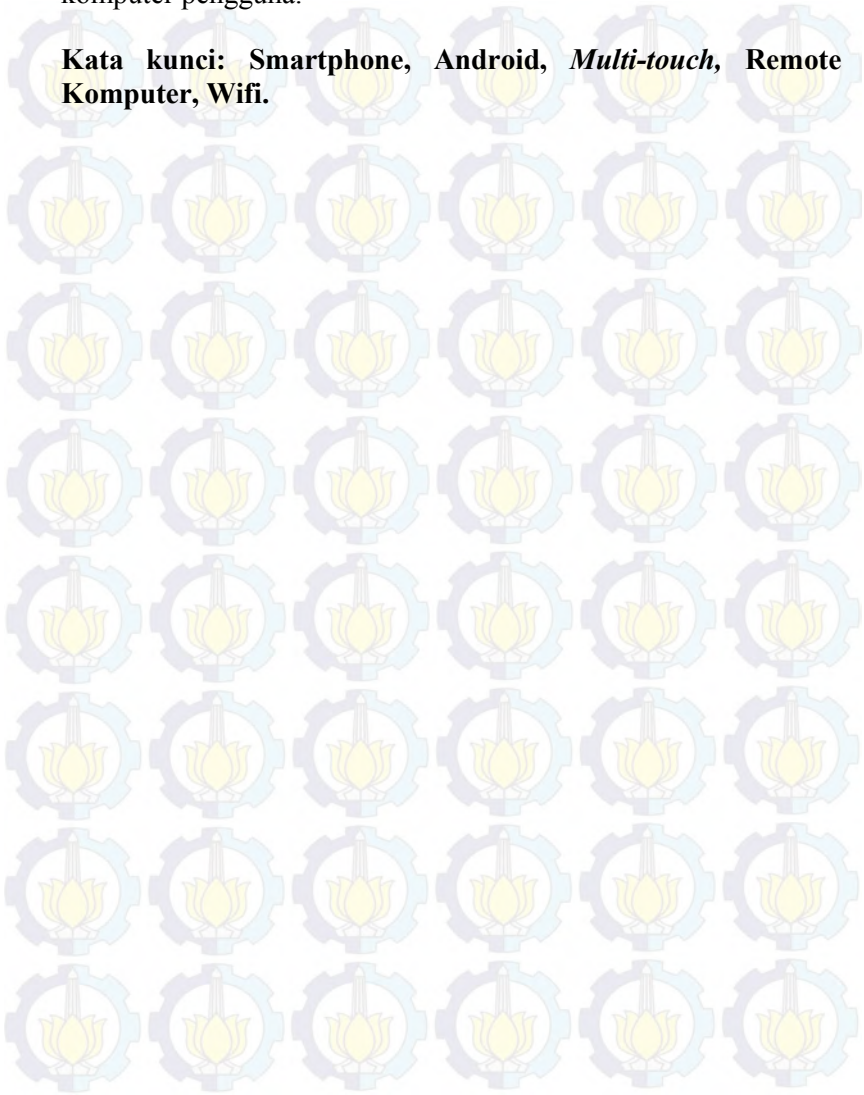
Pada umumnya ponsel Android sudah dilengkapi dengan teknologi Wifi untuk kebutuhan wireless. Dengan memanfaatkan teknologi Wifi dan fitur layar sentuh yang dimiliki oleh semua ponsel Android dapat dibuat sebuah aplikasi pengganti *mouse* atau *touchpad* untuk mengendalikan komputer dari jarak jauh dengan Wifi sebagai penghubungnya.

Dalam Tugas Akhir ini dibuat aplikasi remote komputer dengan pemanfaatan layar sentuh *smartphone* berbasis Android sebagai pengganti *mouse* atau *touchpad* yang dilengkapi dengan fungsi *keyboard* pada komputer. Aplikasi yang dibuat dijalankan pada komputer untuk aplikasi *server* dan ponsel android untuk aplikasi *client*.

Aplikasi ini dapat mempermudah para penggunanya dalam melakukan presentasi dan mengontrol komputernya dari jarak jauh. Serta dengan pemanfaatan username, password, serta MAC

address sehingga dapat mencegah pihak luar mengendalikan komputer pengguna.

Kata kunci: Smartphone, Android, *Multi-touch*, Remote Komputer, Wifi.



KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil‘alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul ***“Aplikasi Remote Komputer Berbasis Android Menggunakan Wifi dengan Multi-touch”***.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
2. Kedua orang tua penulis, Bapak dan ibu yang telah memberikan dukungan jasmani dan rohani, kasih sayang tiada batas, semangat, perhatian, selalu setia dan sabar dalam mendidik dan menghadapi keluh kesah penulis saat mengerjakan Tugas Akhir. Serta do‘a yang luar biasa dan tak terhingga untuk penulis.
3. Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. selaku pembimbing I yang telah memberikan kepercayaan, motivasi dan nasihat yang banyak kepada penulis. Serta membantu dan membimbing penulis dalam menyelesaikan Tugas Akhir ini dengan sabar dan penuh pengertian.

4. Ibu Henning Titi Ciptaningtyas, S.Kom., M.Kom. selaku pembimbing II yang telah membantu dan membimbing penulis dalam menyelesaikan Tugas Akhir ini.
5. Ibu Dr.Eng. Nanik Suciati, S.Kom, M.Kom., selaku ketua jurusan Teknik Informatika ITS sekaligus dosen wali penulis yang telah banyak membantu dan memberi kepercayaan pada penulis dalam menyelesaikan perkuliahan di Teknik Informatika ITS.
6. Segenap Dosen Teknik Informatika ITS yang tidak bisa saya sebutkan satu persatu yang telah membagikan ilmunya kepada penulis selama masa perkuliahan.
7. Pak Yudi, Pak Sugeng dan segenap staf Tata Usaha yang telah memberikan banyak bantuan dan kemudahan pada penulis selama menjalani perkuliahan di Teknik Informatika ITS.
8. Keluarga tercinta, adik dan kakak-kakak penulis yang selalu memberi semangat penulis disaat suka dan duka.
9. Nurul Huda sahabat penulis yang rela membagi ilmunya dan membantu penulis selama mengerjakan Tugas Akhir ini.
10. Muhammad Fatkuri sahabat penulis yang dengan rela selalu mengingatkan penulis untuk menyelesaikan Tugas Akhir.
11. Teman seperjuangan penulis Handito dan Harfie yang selalu saling mengingatkan dan saling membantu dalam menyelesaikan Tugas Akhir.
12. Tidak lupa juga kepada semua pihak yang belum sempat disebutkan satu persatu disini yang telah membantu terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2015

DAFTAR ISI

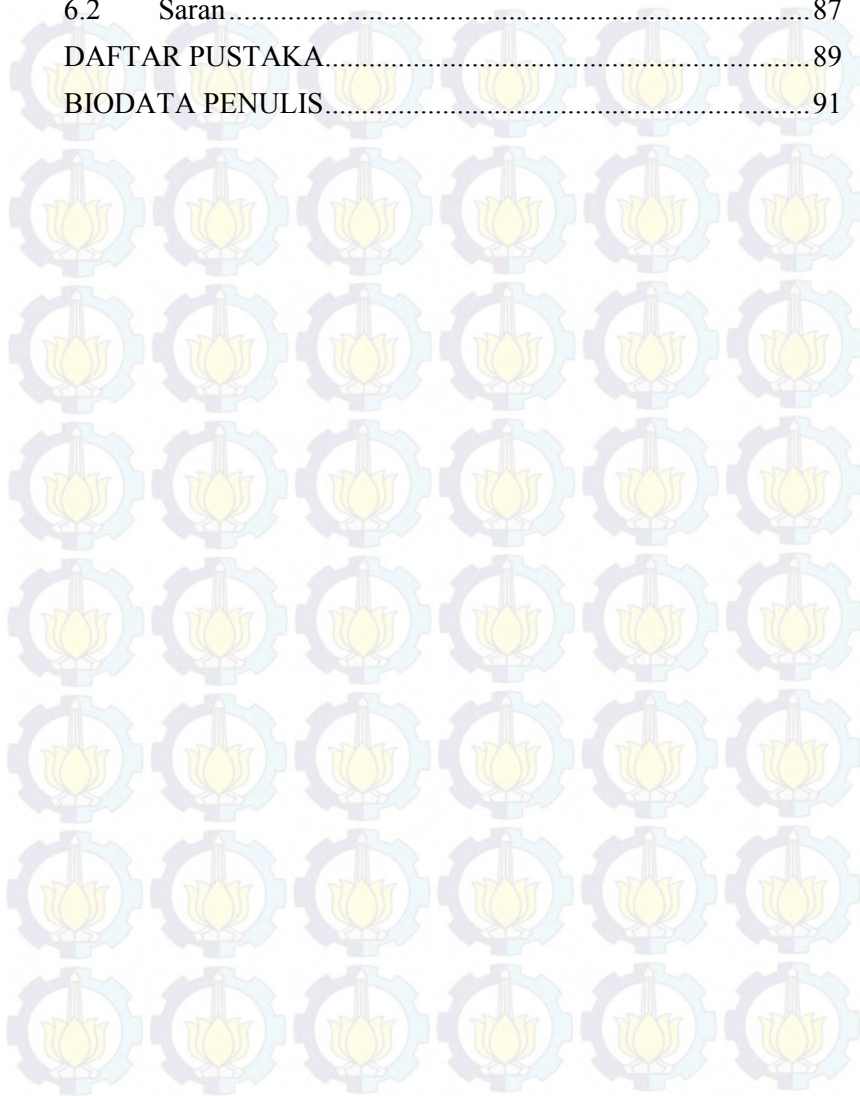
LEMBAR PENGESAHAN.....	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR SOURCE CODE	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	5
BAB II DASAR TEORI.....	7
2.1 Java.....	7
2.1.1 Karakteristik Java	8
2.1.2 Kelebihan dan Kekurangan Java	9
2.1.3 Tahap Kompilasi	10
2.1.4 IDE (Integrated Development Environment)	11
2.2 Android.....	11
2.2.1 Android Activity Lifetime	12

2.2.2	Arsitektur Android.....	13
2.2.3	Fitur-fitur Android.....	17
2.2.4	Keunggulan Android	18
2.2.5	Kekurangan Android	18
2.3	MD5.....	19
2.3.1	Prinsip Dasar MD5	20
2.3.2	Algoritma MD5	20
2.3.3	Pengujian Integritas.....	23
2.3.4	Hash-hash MD5.....	23
2.4	Wifi.....	24
2.4.1	Spesifikasi Wifi	25
2.4.2	Mode akses koneksi Wifi	25
BAB III PERANCANGAN PERANGKAT LUNAK		27
3.1	Arsitektur Umum Sistem.....	27
3.2	Desain Umum Sistem	28
3.3	Perancangan Diagram Kasus	28
3.4	Diagram Alir Aplikasi	30
3.4.1	Diagram Alir Proses Pengisian Username, Password, MAC Address, Port	31
3.4.2	Diagram Alir Proses Pengaturan Mouse Ratio	32
3.4.3	Diagram Alir Proses Penambahan Daftar Log	33
3.4.4	Diagram Alir Proses Menjalankan Server dan Robot.....	34
3.4.5	Diagram Alir Proses Pengisian Username, Password, Server IP, Server Port pada Android	35
3.4.6	Diagram Alir Proses Menjalankan Keyboard.....	36
3.4.7	Diagram Alir Proses Menjalankan Mouse.....	36

3.4.8	Diagram Alir Proses Pembacaan MAC Address pada Android.....	37
3.4.9	Diagram Alir Proses Pembuatan Sesi MD5.....	38
3.5	Perancangan Antarmuka.....	39
3.5.1	Rancangan Antarmuka Account pada Aplikasi Server ..	40
3.5.2	Rancangan Antarmuka Setting pada Aplikasi Server.....	40
3.5.3	Rancangan Antarmuka Log pada Aplikasi Server.....	41
3.5.4	Rancangan Antarmuka Control pada Aplikasi Server....	41
3.5.5	Rancangan Antarmuka Aplikasi Client.....	42
3.5.6	Rancangan Antarmuka Mouse pada Aplikasi Client.....	43
3.5.7	Rancangan Antarmuka Keyboard pada Aplikasi Client.	44
BAB IV IMPLEMENTASI PERANGKAT LUNAK.....		45
4.1	Lingkungan Implementasi	45
4.1.1	Lingkungan Perangkat Lunak.....	45
4.1.2	Lingkungan Perangkat Keras.....	45
4.2	Implementasi Pembuatan Kelas.....	45
4.3	Implementasi Proses Pengisian Username, Password, MAC Address, Port pada Aplikasi Server.....	47
4.4	Implementasi Proses Pengaturan Mouse Ratio.....	49
4.5	Implementasi Proses Penambahan Daftar Log.....	49
4.6	Implementasi Proses Menjalankan Server dan Robot	50
4.7	Implementasi Proses Pengisian Username, Password, Server IP, Server Port pada Android	51
4.8	Implementasi Proses Menjalankan Keyboard	53
4.9	Implementasi proses Multi-touch pada Keyboard.....	57
4.10	Implementasi Proses Menjalankan Mouse	58

4.11	Implementasi Proses Pembacaan MAC Address pada Android.....	61
4.12	Implementasi Proses Pembuatan Sesi MD5	61
4.13	Implementasi Proses Pembuatan Kelas RemoteRobot.java	62
BAB V UJI COBA DAN EVALUASI		65
5.1	Lingkungan Uji Coba	65
5.2	Skenario Uji Coba	65
5.2.1	Uji Coba Fungsionalitas	66
5.2.1.1	Uji Coba Pengisian Username, Password, MAC Address, dan Port.....	66
5.2.1.2	Uji Coba Pengaturan Mouse Ratio	68
5.2.1.3	Uji Coba Penambahan Daftar Log.....	69
5.2.1.4	Uji Coba Menjalankan Server dan Robot.....	71
5.2.1.5	Uji Coba Pengisian Username, Password, Server IP, Server Port pada Android	72
5.2.1.6	Uji Coba Menjalankan Keyboard	74
5.2.1.7	Uji Coba Menjalankan Mouse	76
5.2.1.8	Uji Coba Membaca MAC Address pada Android.....	77
5.2.2	Uji Coba Kebenaran Login, Mouse, Keyboard	78
5.2.2.1	Uji Coba pada Login.....	78
5.2.2.2	Uji Coba Tombol Keyboard pada Android	80
5.2.2.3	Uji Coba Tombol dan Menggerakan Mouse pada Android.....	81
5.2.3	Uji Coba Delay (Latency).....	82
BAB IV KESIMPULAN DAN SARAN.....		87

6.1	Kesimpulan.....	87
6.2	Saran.....	87
	DAFTAR PUSTAKA.....	89
	BIODATA PENULIS.....	91



DAFTAR TABEL

Tabel 2.1 Spesifikasi Wifi	25
Tabel 3.1 Perancangan Diagram Kasus.....	30
Tabel 4.1 Daftar Kelas pada Aplikasi Server	46
Tabel 4.2 Daftar Kelas pada Aplikasi Client.....	47
Tabel 5.1 Uji Coba Pengisian Username, Password, MAC Address, Port	67
Tabel 5.2 Uji Coba Pengaturan Mouse Ratio	68
Tabel 5.3 Uji Coba Penambahan Daftar Log	70
Tabel 5.4 Uji Coba Menjalankan Server dan Robot.....	71
Tabel 5.5 Uji Coba Pengisian Username, Password, Server IP, Server Port pada Android	73
Tabel 5.6 Uji Coba Menjalankan Keyboard.....	75
Tabel 5.7 Uji Coba Menjalankan Mouse.....	76
Tabel 5.8 Uji Coba Membaca MAC Address pada Android.....	78
Tabel 5.9 Uji Coba pada Login	79
Tabel 5.10 Uji Coba Tombol Keyboard pada Android	81
Tabel 5.11 Uji Coba Tombol dan Menggerakkan Mouse pada Android.....	82
Tabel 5.12 Uji Coba Delay.....	85

DAFTAR GAMBAR

Gambar 2.1 Android Activity Lifetime	13
Gambar 2.2 Arsitektur Android.....	14
Gambar 2.3 Satu Operasi MD5	21
Gambar 2.4 Algoritma MD5	22
Gambar 3.1 Arsitektur Umum	27
Gambar 3.2 Diagram Kasus	29
Gambar 3.3 Diagram Alir Proses Pengisian Username, Password, MAC Address, Port	31
Gambar 3.4 Diagram Alir Proses Pengaturan Mouse Ratio	32
Gambar 3.5 Diagram Alir Proses Penambahan Daftar Log	33
Gambar 3.6 Diagram Alir Proses Menjalankan Server dan Robot	34
Gambar 3.7 Diagram Alir Proses Pengisian Username, Password, Server IP, Server Port pada Android	35
Gambar 3.8 Diagram Alir Proses Menjalankan Keyboard	36
Gambar 3.9 Diagram Alir Proses Menjalankan Mouse.....	37
Gambar 3.10 Diagram Alir Proses Pembacaan MAC Address pada Android	38
Gambar 3.11 Diagram Alir Proses Pembuatan Sesi MD5.....	39

Gambar 3.12 Antarmuka Account pada Server	40
Gambar 3.13 Antarmuka Setting pada Server	41
Gambar 3.14 Antarmuka Log pada Server	41
Gambar 3.15 Antarmuka Control pada Server	42
Gambar 3.16 Antarmuka Connect pada Client.....	43
Gambar 3.17 Antarmuka Mouse pada Client	43
Gambar 3.18 Antarmuka Keyboard pada Client	44
Gambar 5.1 Antarmuka Pemasukan Username, Password, MAC Address, dan Server Port pada Server	67
Gambar 5.2 Antarmuka Menu Setting.....	69
Gambar 5.3 Antarmuka Menu Log.....	70
Gambar 5.4 Antarmuka Menu Control.....	72
Gambar 5.5 Antarmuka Pemasukan Username, Password, Server IP, Server Port pada Android.....	74
Gambar 5.6 Antarmuka Menu Keyboard pada Android	75
Gambar 5.7 Antarmuka Menu Mouse pada Android	77

DAFTAR *SOURCE CODE*

Source Code 2.1 Contoh Program Sederhana Menggunakan Bahasa Pemrograman Java	10
Source Code 4.1 Implementasi Pengisian Username, Password, MAC Address, Port pada Server	48
Source Code 4.2 Implementasi Pengaturan Mouse Ratio	49
Source Code 4.3 Implementasi Penambahan Log	50
Source Code 4.4 Implementasi Proses Menjalankan Server dan Robot	51
Source Code 4.5 Implementasi Proses Pengisian Username, Password, Server IP, Server Port pada Android.....	52
Source Code 4.6 Implementasi Proses Menjalankan Keyboard pada Android	55
Source Code 4.7 Implementasi Proses Keyboard pada Server.....	57
Source Code 4.8 Implementasi Proses Multi-touch pada Keyboard	58
Source Code 4.9 Implementasi Mouse Gesture pada Server.....	58
Source Code 4.10 Implementasi Proses Mouse pada Client	60
Source Code 4.11 Implementasi Tombol Mouse pada Server	60
Source Code 4.12 Implementasi Pembacaan MAC Address pada Android.....	61
Source Code 4.13 Implementasi Pembuatan Sesi MD5	62
Source Code 4.14 Potongan Code Kelas RemoteRobot.java	63

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Dalam dunia IT (*Information Technology*) saat sekarang ini, perkembangan teknologi dirasa sangat pesat. Dahulu orang yang mempunyai ponsel masih terbatas dan fitur-fiturnya hanya sekedar untuk mengirim pesan dan menelpon serta hanya kalangan menengah atas saja yang mempunyainya, akan tetapi saat ini hampir semua kalangan mempunyai ponsel dan fiturnya sendiri tidak hanya untuk menelpon tetapi juga bisa sekaligus mengambil gambar, mendengarkan musik, bermain game, dan masih banyak lagi.

Pada saat ini penggunaan sistem operasi pada ponsel bukan merupakan hal yang baru lagi. Dengan adanya sistem operasi pada ponsel, kita dapat memasukkan bermacam-macam aplikasi dibandingkan dengan ponsel yang tanpa sistem operasi. Beberapa tahun yang lalu sistem operasi ponsel yang terkenal adalah Symbian akan tetapi dominasi tersebut seakan runtuh dan digantikan oleh sistem operasi Android.

Android diperkenalkan oleh Google pada tahun 2005 dan didistribusikan pada tahun 2007. Pada umumnya perangkat *mobile* yang menggunakan sistem operasi Android menggunakan layar sentuh dan Android sendiri memang dirancang untuk perangkat *mobile* berbasis *touchscreen*. Perangkat *mobile* yang memakai sistem operasi Android rata-rata sudah terdapat teknologi Wifi didalamnya, dahulu ponsel yang terdapat fitur Wifi merupakan ponsel kelas atas, akan tetapi saat ini ponsel dengan harga dibawah 1 juta sudah bisa merasakan teknologi

Wifi. Wifi sendiri merupakan teknologi untuk menghubungkan antara satu perangkat dengan perangkat lainnya tanpa menggunakan kabel sehingga menjadi lebih praktis.

Laptop zaman sekarang bisa dikatakan pasti memiliki fitur Wifi dan harga Wifi adaptor untuk komputer juga sudah murah, oleh karena itu, penulis mencoba membuat aplikasi yang memanfaatkan teknologi ini untuk menghubungkan antara komputer dengan perangkat *mobile* Android dan dalam hal ini penulis mencoba membuat aplikasi *remote* komputer via Wifi menggunakan Android.

Pada tugas akhir ini, penulis mencoba mempermudah *user* dalam mengendalikan komputernya dari jarak jauh. Hal ini sangat berguna apabila *user* sedang presentasi di suatu ruangan tanpa didampingi seseorang sehingga *user* tidak perlu bolak-balik ke komputer untuk menggerakkan *mouse* ataupun menekan *keyboard*.

1.2 Rumusan Masalah

Terdapat beberapa rumusan masalah yang diangkat pada tugas akhir ini dan akan dipaparkan sebagai berikut:

1. Bagaimana menghubungkan perangkat Android dengan komputer menggunakan Wifi.
2. Bagaimana komputer mengenal perangkat Android sebagai *keyboard* dan *mouse* yang akan digunakan.
3. Bagaimana menghubungkan antara komputer dengan aplikasi ini sehingga tidak terjadi *delay* ketika menggerakkan *mouse* maupun menekan *keyboard* yang ada di layar perangkat Android.
4. Bagaimana menjadikan interface aplikasi dengan fitur *multi-touch* sehingga fungsi-fungsi tombol khusus pada *keyboard* seperti ctrl+a, ctrl+c, ctrl+v, dan lain sebagainya dapat diimplementasikan.

1.3 Batasan Masalah

Permasalahan yang akan dibahas pada tugas akhir ini memiliki beberapa batasan diantaranya:

1. Aplikasi hanya berjalan di sistem operasi Android untuk aplikasi *client* dan Windows untuk aplikasi *server*.
2. Aplikasi hanya untuk menggantikan fungsi *mouse* dan *keyboard* pada komputer.
3. Aplikasi hanya menggunakan Wifi sebagai penghubung antara komputer dengan perangkat Android.
4. Aplikasi menggunakan otentifikasi *username*, *password* serta *MAC address* untuk mengamankan jaringan.
5. Aplikasi memiliki fitur *multi-touch* untuk mengimplementasikan tombol khusus pada *keyboard* seperti ctrl+a, ctrl+c, ctrl+v, dan lain sebagainya.

1.4 Tujuan

Tujuan dari pengerjaan tugas akhir ini adalah untuk mengembangkan kemampuan perangkat *mobile* sehingga dapat digunakan untuk mengontrol komputer dari jarak jauh.

1.5 Manfaat

Beberapa manfaat dari hasil pembuatan tugas akhir ini diantaranya:

1. Dapat mengontrol komputer dari jarak jauh.
2. Mempermudah pengguna dalam melakukan presentasi.
3. Dapat menggantikan *keyboard* dan *mouse* yang ada pada komputer.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1. Penyusunan proposal Tugas Akhir
Penyusunan proposal tugas akhir merupakan tahap awal dalam proses pengerjaan tugas akhir. Di dalam proposal

dijelaskan usulan tugas akhir mengenai Aplikasi Remote Komputer Berbasis Android Menggunakan WiFi Dengan *Multi-touch*.

2. Studi Literatur

Pada tahap ini dilakukan pemahaman literatur yang berhubungan dengan Java, Android dan Wifi. Literatur yang dipelajari meliputi bahasa pemrograman Java, pemrograman Android, dan algoritma MD5 untuk pembuatan *session code*.

3. Perancangan Perangkat Lunak

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan rancangan sistem yang akan dibuat pada karya tugas akhir ini, mulai dengan deskripsi umum mengenai aplikasi yang akan dibuat, perancangan proses-proses yang ada, serta alur-alur prosesnya.

4. Implementasi Perangkat Lunak

Pada tahap ini akan dilakukan implementasi rancangan sistem yang telah dibuat. Tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya sehingga menjadi sebuah aplikasi yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan Evaluasi

Pada tahap ini akan dilakukan pengujian pada perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Dengan melakukan pengujian dan evaluasi dimaksudkan juga untuk mengevaluasi jalannya aplikasi, mencari masalah yang mungkin timbul dan melakukan perbaikan jika terdapat kekurangan.

6. Penyusunan Laporan Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat

lunak yang telah dibuat. Laporan Tugas Akhir ini bertujuan untuk memberikan gambaran dari pengerjaan Tugas Akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

1.7 Sistematika Penulisan

Pada tahap ini dilakukan penyusunan laporan yang akan dibagi menjadi beberapa bab sebagai berikut:

Bab I. Pendahuluan

Bab ini berisi latar belakang masalah, tujuan, manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan.

Bab II. Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III. Perancangan Perangkat Lunak

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan rancangan sistem yang akan dibuat pada karya tugas akhir ini, mulai dengan deskripsi umum mengenai aplikasi yang akan dibuat, perancangan proses-proses yang ada, serta diagram alir prosesnya, dan antarmuka perangkat lunak.

Bab IV. Implementasi Perangkat Lunak

Bab ini membahas implementasi dari desain sistem yang dilakukan pada tahap desain, disertai dengan potongan kode program dalam perangkat lunak.

Bab V. Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang berisi kesimpulan dari hasil uji coba yang dilakukan serta saran untuk pengembangan perangkat lunak selanjutnya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

BAB II

DASAR TEORI

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian perangkat lunak. Literatur yang akan dijelaskan meliputi tentang Java, Android, Wifi, MD5, dan bahasa pemrograman Java yang akan digunakan dalam Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Java

Java adalah bahasa pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Mereka lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional.

Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *bytecode* dan dapat dijalankan pada berbagai JVM (*Java Virtual Machine*). Java merupakan bahasa pemrograman yang bersifat umum, dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa sistem

operasi yang berbeda, sehingga java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*".

2.1.1 Karakteristik Java

Beberapa karakteristik Java antara lain :

- Bahasa pemrograman Java dikenal sederhana (simple), karena menggunakan Sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan multiple inheritance. Java juga menggunakan automatic memory allocation dan memory garbage collection.
- Java menggunakan pemrograman berorientasi objek *Object Oriented Programming* (OOP) yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.
- Java dapat didistribusikan dengan mudah. Java digunakan untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada Java.
- Program Java dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada platform yang berbeda-beda. Ini yang menjadi kekuatan Java, berbeda dengan VB yang hanya berjalan di windows atau PHP yang hanya berjalan pada Apache.
- Java mempunyai reliabilitas yang tinggi. *Compiler* pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception* handling untuk membantu mengatasi error pada pemrograman.
- Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem

komputer yang menjalankan aplikasi tersebut.

- Program Java merupakan platform independent. Program cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* berbeda dengan *Java Virtual Machine*.
- Java bersifat portable. *Source code* maupun program Java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.
- Performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers* (JIT).
- Java mempunyai kemampuan *Multithread*. Java mampu membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.
- Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

2.1.2 Kelebihan dan Kekurangan Java

- Multiplatform. Java dapat dijalankan di beberapa sistem operasi seperti Windows, Linux, Mac OS, dan Sun Solaris, sehingga memungkinkan program yang dibuat di Linux dijalankan di dalam Windows.
- OOP (Object Oriented Programming - Pemrograman berorientasi objek).
- Perpustakaan kelas yang lengkap, Java terkenal dengan kelengkapan *library* yang sangat memudahkan para pemrogram untuk membangun aplikasinya.
- Bergaya C++, memiliki sintaksis seperti bahasa pemrograman C++ sehingga menarik minat pemrogram C++ untuk pindah ke Java.
- Pengumpulan sampah otomatis, memiliki fasilitas

pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung.

- Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan *platform* lain. Untuk J2SE, misalnya SWT-AWT yang sampai sekarang tidak berfungsi pada Mac OS X.
- Mudah didekompilasi. Dekompilasi adalah proses mengembalikan dari kode jadi menjadi kode sumber. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak.
- Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal).

```
// Outputs "Hello, world!" and then exits
```

```
public class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello, world!");
    }
}
```

Source Code 2.1 Contoh Program Sederhana Menggunakan Bahasa Pemrograman Java

2.1.3 Tahap Kompilasi

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '.java'.
2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '.class'.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.

4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin.

2.1.4 IDE (*Integrated Development Environment*)

Banyak pihak telah membuat IDE untuk Java. Beberapa yang populer antara lain:

- Dr. Java, program gratis yang dikembangkan oleh Universitas Rice, Amerika Serikat.
- BlueJ, program gratis yang dikembangkan oleh Universitas Monash, Australia.
- NetBeans (open source- Common Development and Distribution License (CDDL)). NetBeans disponsori Sun Microsystems, dan versi terkininya memiliki Matisse, sebuah GUI Editor yang menurut pendapat umum merupakan yang terbaik.
- Eclipse JDT (open source- Eclipse Public License). Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse Foundation' (beserta individu-individu lain). Banyak nama besar yang ikut dalam 'Eclipse Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.
- IntelliJ IDEA (commercial, free 30-day trial).
- Oracle JDeveloper (free).
- Xinox JCreator (ada versi berbayar maupun free).
- JCreator ditulis dalam C/C++ sehingga lebih cepat dan menggunakan memori lebih sedikit.

2.2 Android

Android adalah sistem operasi untuk *mobile device* yang awalnya dikembangkan oleh Android Inc. Google Inc membeli

Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel. Pada saat perilis perdana Android, 5 November 2007, Android bersama *open handset alliance*, yaitu konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia, menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, google merilis kode-kode android di bawah lisensi apache. Android dibuat berdasarkan kernel Linux yang dimodifikasi. Aplikasi Android ditulis dengan bahasa Java, menggunakan Java Core Libraries. Aplikasi Android dijalankan di atas VM bernama Dalvik Virtual Machine.

Pada saat ini, sudah banyak vendor smartphone yang memproduksi berbasis android, hal ini terjadi karena android adalah sistem operasi yang open source sehingga bebas di distribusikan & dipakai oleh vendor manapun. Android itu sendiri sangat lengkap baik dari segi sistem operasi, aplikasi dan tool pengembangan, market aplikasi serta dukungan yang sangat tinggi dari komunitas open source didunia.

Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

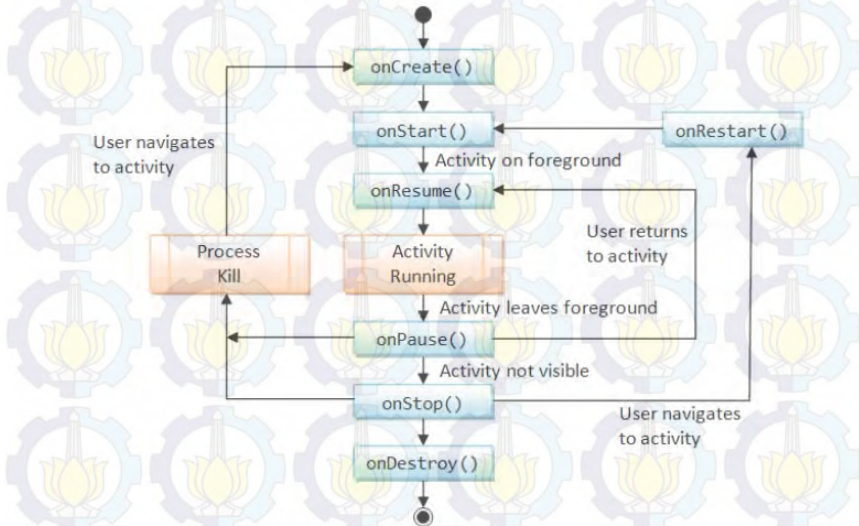
2.2.1 Android Activity Lifetime

Sebuah Activity biasanya ditampilkan dalam satu screen sebagai bagian komponen dari View (yang terdiri dari komponen UI, widget atau control), yang bertugas melakukan interaksi dengan user untuk melakukan sebuah task tunggal (misalnya, melihat informasi, mengedit data, atau memasukkan data).

Sebuah aplikasi Android biasanya memiliki satu atau lebih Activity. Salah satu Activity ditandai sebagai startup activity, yang pada gilirannya memulai activity berikutnya, melalui sebuah intent. Untuk membuat suatu activity, kita

melakukan `extend` terhadap class `android.app.Activity`, dan melakukan `override` beberapa method, khususnya method `OnCreate()`.

Kita bisa mengatur konten view dari activity secara bahasa pemrograman java atau juga bisa melalui kode layout file XML.



Gambar 2.1 Android Activity Lifetime

Daur hidup sebuah activity diatur melalui method call-back, yang di definisikan pada class dasar Activity pada gambar diatas.

2.2.2 Arsitektur Android

Google sebagai pencipta Android yang kemudian diasuh oleh *Open Handset Alliance* mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Tumpukan paling bawah adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk

membangun Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Bertempat di level yang sama dengan library adalah lapisan runtime yang mencakup serangkaian inti library Java. Dengannya, para programmer dapat mengembangkan aplikasi untuk Android menggunakan bahasa pemrograman Java. Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi- fungsi dasar smartphone.



Gambar 2.2 Arsitektur Android

1. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux

menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

- Media Library untuk memutar dan merekam berbagai macam format audio dan video.
- Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi.
- Graphic Library termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
- SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
- SSL dan WebKit untuk browser dan keamanan internet.

3. Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya:

- Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java

serta beberapa pustaka khusus android.

- Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format Dalvik Executable (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalokasian memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti threading dan manajemen memori yang terbatas.

4. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

- Activity Manager, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "Backstack" untuk navigasi penggunaan.
- Content Providers, berfungsi untuk merangkul data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
- Resource Manager, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
- Location Manager, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
- Notification Manager, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain

sebagainya yang akan ditampilkan pada status bar.

5. Application Layer

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan widget. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android runtime dengan menggunakan kelas dan service yang tersedia pada framework aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (native) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.2.3 Fitur-fitur Android

Fitur-fitur yang dimiliki android sebagai berikut:

- Kerangka aplikasi: memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat telepon seluler.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- Mendukung teknologi konektivitas: GSM, Bluetooth, EDGE, 3G, 4G dan WiFi (tergantung piranti keras).
- Mendukung hardware tambahan: Kamera, Global Positioning System (GPS), kompas, NFC dan

accelerometer (tergantung piranti keras).

- Multi-touch: android sudah mendukung multi-touch. Ini adalah kemampuan perangkat untuk mendeteksi beberapa sentuhan pada layar pada satu waktu. Fitur ini awalnya dinonaktifkan pada tingkat kernel karena memungkinkan pelanggaran hak paten pada teknologi layar sentuh Apple.

2.2.4 Keunggulan Android

a. Lengkap (*Complete Platform*)

Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang mengembangkan aplikasi.

b. Terbuka (*Open Source Platform*)

Platform android disediakan melalui lisesnsi open source. Pengembang dapat dengan bebas mengembangkan aplikasi. Android sendiri menggunakan linux kernel 2.6.

c. Gratis (*Free Platform*)

Tidak ada lisensi atau biaya royalti untuk pengembangan pada aplikasi android. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.2.5 Kekurangan Android

Diantara kekurangan-kekurangan android adalah sebagai berikut:

- Koneksi Internet yang terus menerus. Dengan koneksi internet yang terus menerus maka handphone jadi boros baterai sehingga akan sering melakukan *charge* pada handphone, tetapi untuk versi Android 2.2 ke atas koneksi internet bisa di setting dengan mengaktifkan atau menonaktifkan nya.
- Dengan tidak mengaktifkan koneksi internetnya maka

HP menjadi seperti HP biasa yang hanya bias untuk mengirim/menerima pesan dan menelepon serta di telpon saja.

- Iklan. Setiap menjalankan sebuah aplikasi maka akan muncul berbagai macam iklan apalagi dengan mengaktifkan koneksi internet nya.
- Boros baterai karena setiap aplikasi yang di jalankan, sinyal, RUM yang digunakan akan menggunakan daya baterai. Apalagi sinyal di Indonesia yang belum stabil maka baterai akan mudah habis karena mempertahankan sinyal agar tetap stabil.

2.3 MD5

MD5 merupakan fungsi hash satu arah yang didesain oleh Ronald Rivest dengan *hash value* 128 bit. Pada standard internet, MD5 telah dimanfaatkan pada bermacam-macam aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah file.

Algoritma MD5 secara garis besar adalah mengambil pesan yang mempunyai panjang variabel diubah menjadi „sidik jari“ atau „intisari pesan“ yang mempunyai panjang tetap yaitu 128 bit. „Sidik jari“ ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari „sidik jari“ MD5.

Message Digest 5 (MD5) adalah salah satu dari serangkaian algoritma *Message Digest* yang didesain oleh Professor Ronald Rivest dari MIT. Saat kerja analitik menunjukkan bahwa pendahulu MD5 -MD4- mulai tidak aman, MD5 kemudian di desain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

MD5 banyak digunakan pada bermacam-macam aplikasi termasuk SSL/TLS, IPsec dan protocol-protokol kriptografi lainnya. MD5 juga biasa digunakan pada implementasi

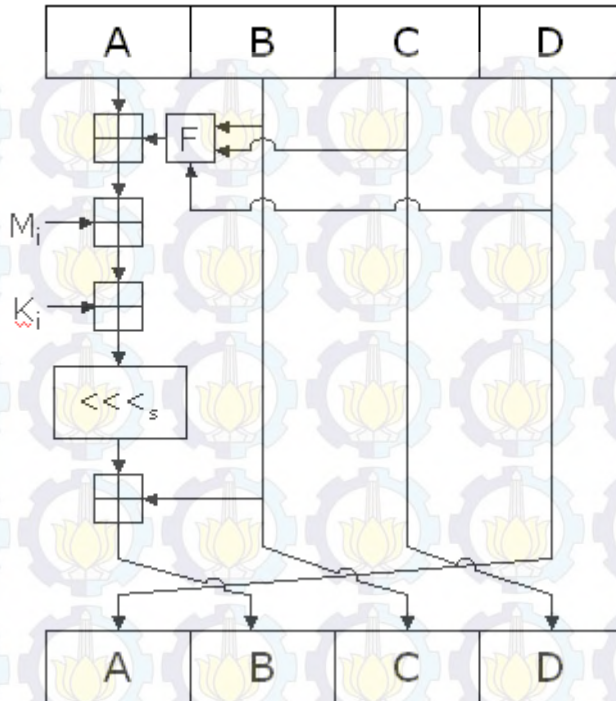
Timestamping Mechanism, Commitment Schemes, dan aplikasi pengecekan integritas pada *online software*.

2.3.1 Prinsip Dasar MD5

Message Digest 5 (MD5) adalah salah satu penggunaan fungsi *hash* satu arah yang paling banyak digunakan. MD5 merupakan fungsi *hash* kelima yang dirancang oleh Ron Rivest dan didefinisikan pada RFC 1321. MD5 merupakan pengembangan dari MD4 dimana terjadi penambahan satu ronde. MD5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Keluaran dari MD5 berupa 4 buah blok yang masing-masing 32 bit yang mana akan menjadi 128 bit yang biasa disebut nilai *hash*.

2.3.2 Algoritma MD5

MD5 terdiri atas 64 operasi, dikelompokkan dalam empat putaran dari 16 operasi. F adalah fungsi nonlinear; satu fungsi digunakan pada tiap-tiap putaran. M_i menunjukkan blok 32 bit dari masukan pesan, dan K_i menunjukkan konstanta 32 bit, berbeda untuk tiap-tiap operasi.



Gambar 2.3 Satu Operasi MD5

Pada gambar 2.3 \lll_s menunjukkan perputaran bit kiri oleh s , s bervariasi untuk tiap-tiap operasi. menunjukkan tambahan modulo 2^{32} . MD5 memproses variasi panjang pesan ke dalam keluaran 128 bit dengan panjang yang tetap. Pesan masukan dipecah menjadi dua gumpalan blok 512 bit. Pesan ditata sehingga panjang pesan dapat dibagi 512. Penataan bekerja sebagai berikut : bit tunggal pertama ,1, diletakkan pada akhir pesan. Proses ini diikuti dengan serangkaian nol (0) yang diperlukan agar panjang pesan lebih dari 64 bit dan kurang dari kelipatan 512. Bit-bit sisa diisi dengan 64 bit integer untuk menunjukkan panjang pesan yang asli. Sebuah pesan selalu ditata setidaknya dengan 1 bit tunggal, seperti jika panjang pesan

adalah kelipatan 512 dikurangi 64-bit untuk informasi panjang (panjang mod(512) = 448), sebuah blok baru dari 512-bit ditambahkan dengan 1 bit diikuti dengan 447 bit-bit nol (0) diikuti dengan panjang 64 bit.

Algoritma MD5 yang utama beroperasi pada kondisi 128 bit, dibagi menjadi empat 32 bit kata, dilambangkan A, B, C, dan D. Operasi tersebut diinisialisasi dijaga untuk tetap konstan. Register A,B,C dan D diinisialisasi dengan bilangan hexadecimal.

kata A: 01 23 45 67

kata B: 89 AB CD EF

kata C: FE DC BA 98

kata D: 76 54 32 10

Register-register ini biasa disebut dengan nama *chain variable* atau variabel rantai.

Algoritma utama kemudian beroperasi pada masing-masing blok pesan 512 bit, masingmasing blok melakukan pengubahan terhadap kondisi. Pemrosesan blok pesan terdiri atas empat tahap, batasan putaran, tiap putaran membuat 16 operasi serupa berdasar pada fungsi non-linear F, tambahan modular, dan rotasi ke kiri. Gambar 2.4 mengilustrasikan satu operasi dalam putaran. Ada empat macam kemungkinan fungsi F, berbeda dari yang digunakan pada tiap-tiap putaran.

Masing-masing fungsi pada Gambar 2.4 \wedge , \vee , \neg , \oplus menunjukkan operasi logika AND, OR, NOT, dan XOR.

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X \vee \neg Z)$$

Gambar 2.4 Algoritma MD5

2.3.3 Pengujian Integritas

MD5 digunakan secara luas dalam dunia perangkat lunak untuk menyediakan semacam jaminan bahwa file yang diambil (*download*) belum terdapat perubahan. Seorang *user* dapat membandingkan MD5 *sum* yang dipublikasikan dengan *checksum* dari file yang diambil. Dengan asumsi bahwa *checksum* yang dipublikasikan dapat dipercaya akan keasliannya, seorang *user* dapat secara yakin bahwa file tersebut adalah file yang sama dengan file yang dirilis oleh para *developer*, jaminan perlindungan dari *Trojan Horse* dan virus komputer yang ditambahkan pada perangkat lunak. Bagaimanapun juga, seringkali kasus yang terjadi bahwa *checksum* yang dipublikasikan tidak dapat dipercaya (sebagai contoh, *checksum* didapat dari channel atau lokasi yang sama dengan tempat mengambil file), dalam hal ini MD5 hanya mampu melakukan *error-checking*. MD5 akan mengenali file yang didownload tidak sempurna, cacat atau tidak lengkap.

2.3.4 Hash-hash MD5

Hash-hash MD5 sepanjang 128 bit (16 byte), yang dikenal juga sebagai ringkasan pesan, secara tipikal ditampilkan dalam bilangan *heksadesimal* 32 digit. Berikut ini merupakan contoh pesan ASCII sepanjang 43 byte sebagai masukan dan *hash* MD5 terkait :

MD5("The quick brown fox jumps over the lazy dog") = 9e107d9d372bb6826bd81d3542a419d6.

Bahkan perubahan yang kecil pada pesan akan (dengan probabilitas lebih) menghasilkan hash yang benar-benar berbeda, misalnya pada kata "dog", huruf d diganti menjadi c :

MD5("The quick brown fox jumps over the lazy cog") = 1055d3e698d289f2af8663725127bd4b.

Hash dari panjang nol ialah:

MD5("")=d41d8cd98f00b204e9800998ecf8427e

2.4 Wifi

WiFi adalah singkatan dari *Wireless Fidelity* (sering ditulis dengan Wi-fi, WiFi, Wifi, wifi). Pengertian Wi-Fi adalah koneksi tanpa kabel yang menghubungkan jaringan komputer, seperti ponsel yang mempergunakan teknologi radio sehingga pengguna dapat melakukan transfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel baik di rumah maupun di kantor-kantor dan pusat-pusat bisnis. Oleh Karena itu banyak orang mengasosiasikan Wi-Fi dengan “Kebebasan” karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang meeting, kamar hotel, kampus, dan café-café yang bertanda “Wi-Fi Hot Spot”. Juga salah satu kelebihan dari Wi-Fi adalah kecepatannya yang beberapa kali lebih cepat dari modem kabel yang tercepat. Jadi pemakai Wi-Fi tidak lagi harus berada di dalam ruang kantor untuk bekerja.

Wi-Fi *Alliance* mendefinisikan Wi-Fi sebagai "produk jaringan wilayah lokal nirkabel (WLAN) apapun yang didasarkan pada standar *Institute of Electrical and Electronics Engineers* (IEEE) 802.11". Meski begitu, karena kebanyakan WLAN zaman sekarang didasarkan pada standar tersebut, istilah "Wi-Fi" dipakai dalam bahasa Inggris umum sebagai sinonim "WLAN".

Wifi mempunyai sejarah keamanan yang berubah. Sistem enkripsi pertamanya, WEP, terbukti mudah ditembus. Protokol berkualitas lebih tinggi lagi, WPA dan WPA2 kemudian ditambahkan, tetapi sebuah fitur opsional yang ditambahkan tahun 2007 bernama Wi-Fi Protected Setup (WPS), memiliki celah yang memungkinkan penyerang mendapatkan kata sandi WPA atau WPA2 router dari jarak jauh dalam beberapa jam saja. Sejumlah perusahaan menyarankan untuk mematikan fitur WPS. Wi-Fi Alliance sejak itu memperbarui rencana pengujian dan program sertifikasinya untuk menjamin semua peralatan yang baru disertifikasi kebal dari serangan AP PIN yang keras.

2.4.1 Spesifikasi Wifi

Wifi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11 seperti pada Tabel 2.1. Hardware Wifi yang ada di pasaran saat ini berupa PCI, USB, PCMCIA, dan Compact Flash.

Tabel 2.1 Spesifikasi Wifi

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	2,4 GHz	b
802.11a	54 Mb/s	5 GHz	a
802.11g	54 Mb/s	2,4 GHz	b, g
802.11n	100 Mb/s	2,4 GHz	b, g, n

2.4.2 Mode akses koneksi Wifi

Ada 2 mode akses koneksi Wifi, yaitu:

a. Ad-Hoc

Jenis koneksi ini adalah jenis dimana beberapa komputer terhubung secara langsung. Keuntungannya, lebih murah dan praktis bila yang terkoneksi hanya 2 atau 3 komputer, tanpa harus membeli *access point*.

b. Infrastruktur

Menggunakan *access point* yang berfungsi sebagai pengatur lalu lintas data, sehingga memungkinkan banyak klien dapat saling terhubung antara satu sama lainnya melalui jaringan (*network*).

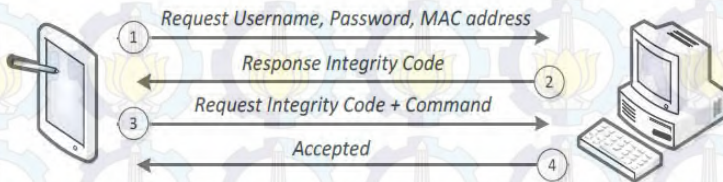
BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan rancangan sistem yang akan dibuat pada karya tugas akhir ini, mulai dengan deskripsi umum mengenai aplikasi yang akan dibuat, perancangan proses-proses yang ada, serta diagram alir prosesnya, dan antarmuka perangkat lunak.

3.1 Arsitektur Umum Sistem

Rancangan arsitektur dari sistem yang akan dibuat dapat digambarkan seperti pada Gambar 3.1. Proses pertama pada Gambar 3.1 yaitu, pengguna menghubungkan perangkat Android dengan komputer yang akan dikendalikan menggunakan Wifi, pada aplikasi komputer pengguna memasukkan *username*, *password*, *server port*, dan *MAC address* perangkat Android, pengguna juga bisa mengatur *mouse ratio*, lalu pengguna menjalankan *server* dan *robot*. Setelah itu, pengguna *login* di Android dengan memasukkan *username*, *password*, *server IP*, dan *server port*. Setelah *input* yang dimasukan sudah benar maka komputer akan mengirimkan kode integritas ke Android dan sesi terbentuk. Dengan kode integritas tersebut Android mengirim perintah ke komputer dan perintah dijalankan. Perintah yang dijalankan dapat dilihat pada daftar *log*.



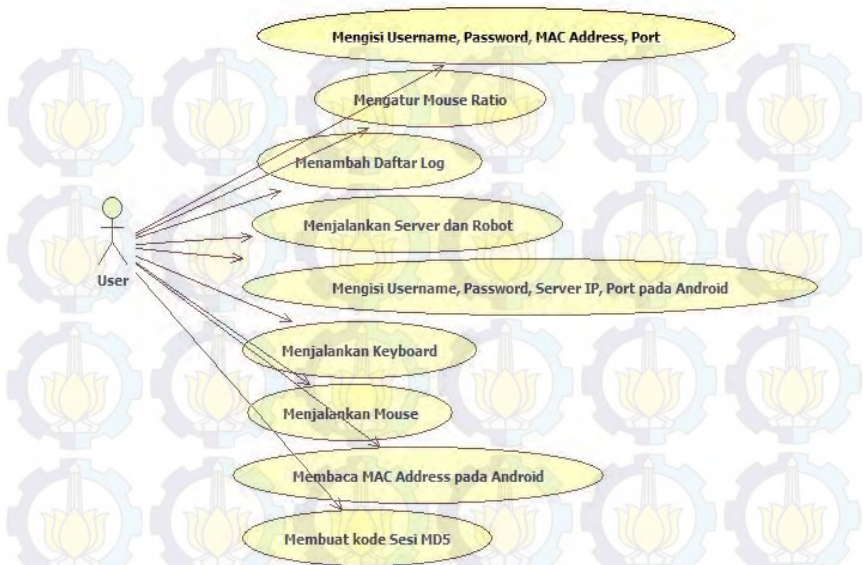
Gambar 3.1 Arsitektur Umum

3.2 Desain Umum Sistem

Dalam Tugas Akhir ini akan membangun suatu perangkat lunak pada komputer dan Android. Perangkat lunak yang merupakan sebuah aplikasi berbasis Android yang berfungsi sebagai alat untuk mengendalikan *keyboard* dan *mouse* yang ada di komputer. Aplikasi yang dibuat pada perangkat lunak ini menggunakan bahasa pemrograman Java.

3.3 Perancangan Diagram Kasus

Perancangan diagram kasus akan menggambarkan fungsionalitas sistem beserta aktor yang terlibat. Pada sistem ini hanya terdapat satu aktor yaitu pengguna seperti pada Gambar 3.2. Setelah menjalankan aplikasi *Server* pada komputer, pengguna diwajibkan mengisi *username*, *password*, *MAC address*, dan *port* yang ada pada menu *account*, pengguna juga bisa mengatur *mouse ratio* yang ada pada menu *setting* lalu menjalankan *server* dan *robot* yang ada pada menu *control*. Pada aplikasi Android pengguna memasukkan *username*, *password*, *server IP*, dan *server port* yang sama dengan aplikasi di komputer. Setelah koneksi berhasil, pengguna dapat menjalankan *keyboard* maupun *mouse* sesuai pilihan.



Gambar 3.2 Diagram Kasus

Dari perancangan diagram kasus dibuatlah table diagram kasus seperti pada tabel 3.1 yang akan menjelaskan mengenai diagram kasus yang ada pada perangkat lunak.

Tabel 3.1 Perancangan Diagram Kasus

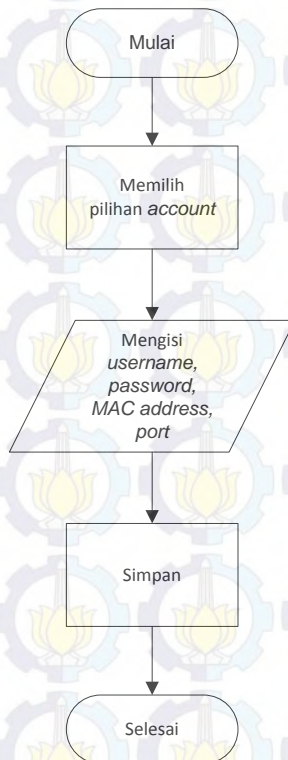
No	Kode Diagram Kasus	Nama Diagram Kasus	Keterangan
1	UJI-01	Mengisi Username, Password, MAC Address, dan Port	Proses pengisian username, password, MAC address, dan port
2	UJI-02	Mengatur Mouse Ratio	Proses pengaturan mouse ratio
3	UJI-03	Menambah Daftar Log	Proses penambahan dan penghapusan log
4	UJI-04	Menjalankan Server dan Robot	Proses menjalankan server dan robot
5	UJI-05	Mengisi Username, Password, Server IP, Server Port pada Android	Proses pengisian username, password, server IP, server port pada Android
6	UJI-06	Menjalankan Keyboard	Proses menjalankan keyboard
7	UJI-07	Menjalankan Mouse	Proses menjalankan mouse
8	UJI-08	Membaca MAC Address pada Android	Proses pembacaan MAC address
9	UJI-09	Membuat Sesi MD5	Proses pembuatan sesi MD5

3.4 Diagram Alir Aplikasi

Alur setiap proses yang terdapat pada perangkat lunak digambarkan pada diagram alir untuk mempermudah pemahaman secara garis besar proses yang ada pada sistem.

3.4.1 Diagram Alir Proses Pengisian *Username, Password, MAC Address, Port*

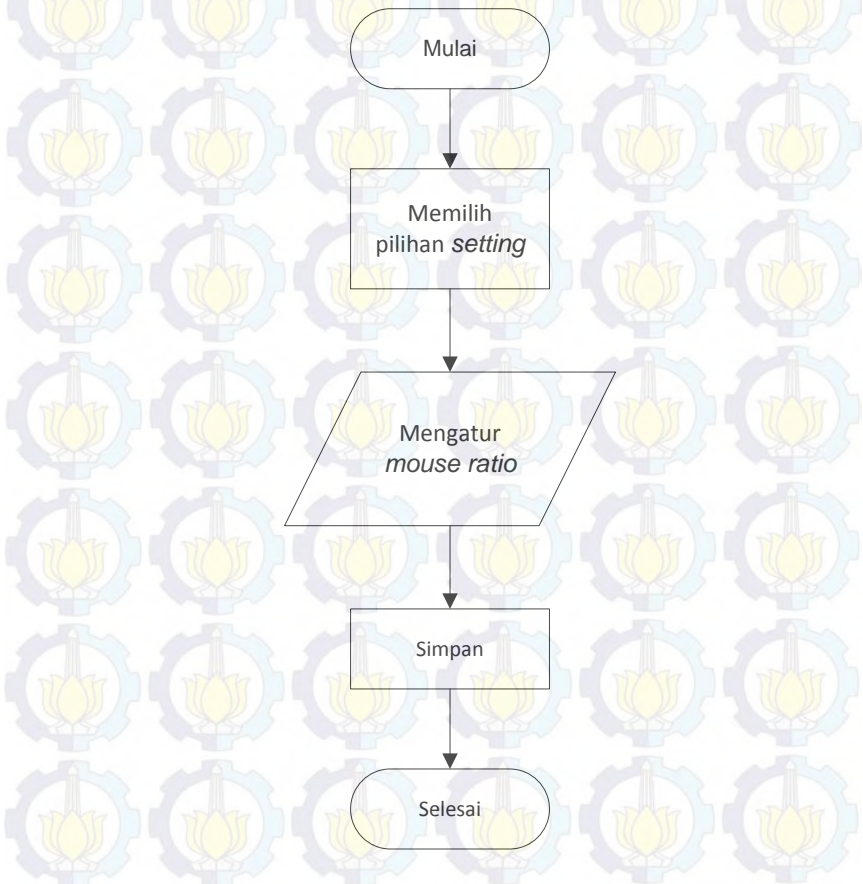
Pada Gambar 3.3 disajikan diagram alir proses pengisian *username, password, MAC address, dan port*. Proses diawali dengan memilih menu *account* pada halaman utama. Kemudian akan tampil *form account*. *User* memasukkan *username, password, MAC address, dan port*. Data tersebut akan disimpan di dalam data *account*.



Gambar 3.3 Diagram Alir Proses Pengisian *Username, Password, MAC Address, Port*

3.4.2 Diagram Alir Proses Pengaturan *Mouse Ratio*

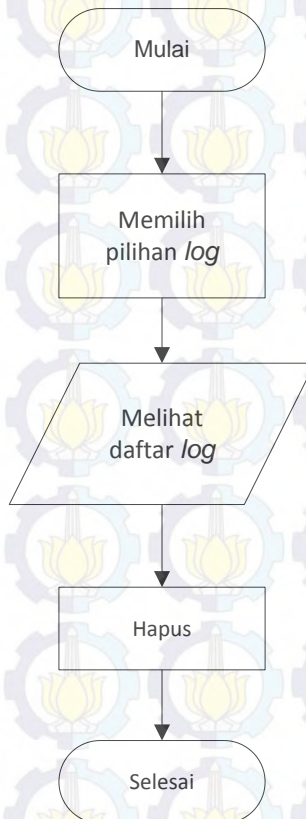
Pada Gambar 3.4 disajikan diagram alir proses pengaturan *mouse ratio*. Proses diawali dengan memilih pilihan *setting*. Kemudian akan tampil *form combo box*. *User* memilih *ratio* yang diinginkan. Setelah itu, *user* menyimpan pilihan tersebut.



Gambar 3.4 Diagram Alir Proses Pengaturan *Mouse Ratio*

3.4.3 Diagram Alir Proses Penambahan Daftar Log

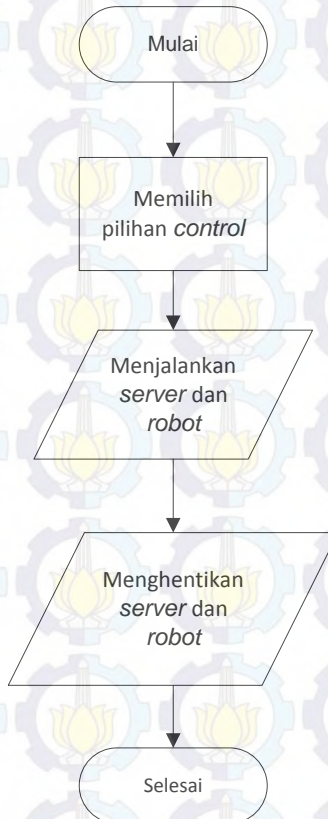
Pada Gambar 3.5 disajikan diagram alir proses penambahan dan penghapusan *log*. Proses diawali setelah aplikasi berjalan lalu *user* memilih pilihan *log*. Kemudian akan tampil *form server* dan *robot*, *user* juga bisa menghapus *log* tersebut dengan memilih pilihan *clear*.



Gambar 3.5 Diagram Alir Proses Penambahan Daftar Log

3.4.4 Diagram Alir Proses Menjalankan *Server* dan *Robot*

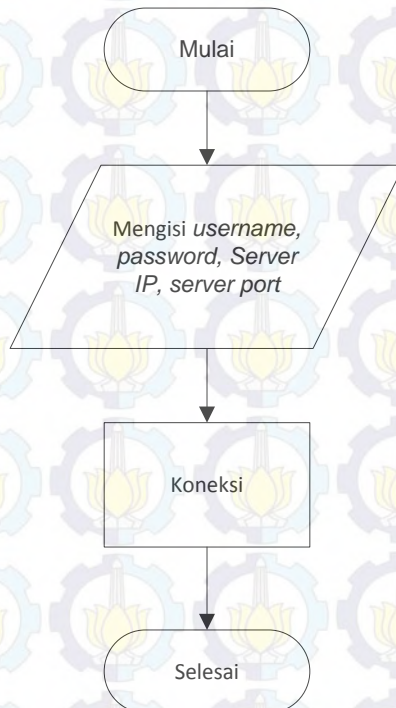
Pada Gambar 3.6 disajikan diagram alir proses menjalankan *server* dan *robot*. Proses diawali dengan memilih menu *control* lalu memilih *run* yang ada pada *server* dan *robot*. *User* juga bisa menghentikan atau me-restart proses tersebut.



Gambar 3.6 Diagram Alir Proses Menjalankan *Server* dan *Robot*

3.4.5 Diagram Alir Proses Pengisian *Username, Password, Server IP, Server Port* pada Android

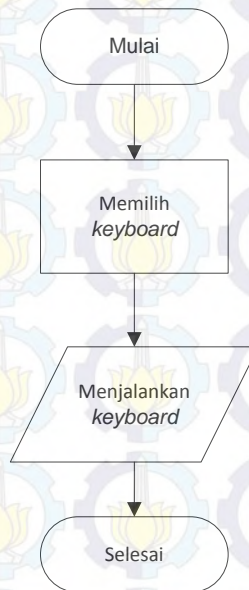
Pada Gambar 3.7 disajikan diagram alir proses pengisian *username, password, server IP, server port* pada Android. Proses diawali dengan memasukkan *username, password, server IP, server port* yang sesuai pada aplikasi di Android. Kemudian *user* memilih *connect* untuk menghubungkan antara perangkat Android dengan *server*.



Gambar 3.7 Diagram Alir Proses Pengisian *Username, Password, Server IP, Server Port* pada Android

3.4.6 Diagram Alir Proses Menjalankan *Keyboard*

Pada Gambar 3.8 disajikan diagram alir proses menjalankan *keyboard*. Proses baru bisa berlangsung setelah *server* dan perangkat Android telah terhubung satu sama lain, setelah itu *user* memilih menu *keyboard* yang ada di aplikasi Android. *User* dapat memilih pilihan tombol *keyboard* yang ada pada aplikasi tersebut.

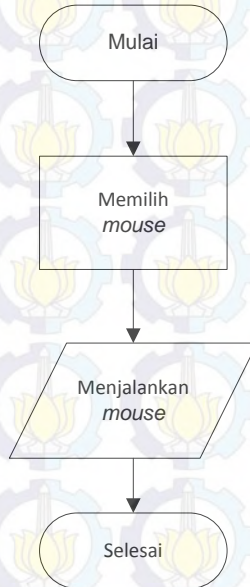


Gambar 3.8 Diagram Alir Proses Menjalankan *Keyboard*

3.4.7 Diagram Alir Proses Menjalankan *Mouse*

Pada Gambar 3.9 disajikan diagram alir proses menjalankan *mouse*. Proses baru bisa berlangsung setelah *server* dan perangkat Android telah terhubung satu sama lain, setelah itu

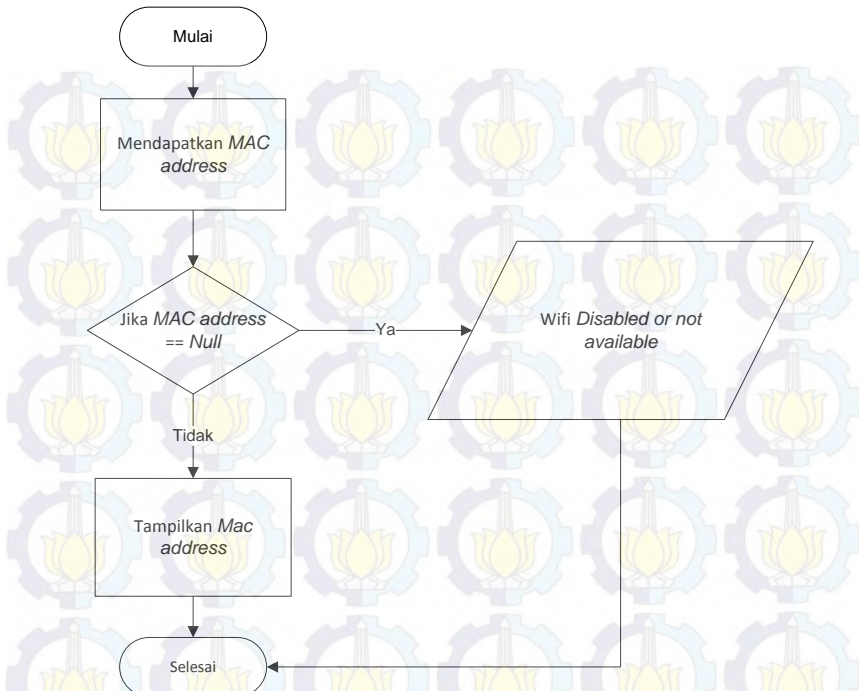
user memilih menu *mouse* yang ada di aplikasi Android. *User* dapat memilih tombol atau menggerakkan *mouse* yang ada pada aplikasi tersebut.



Gambar 3.9 Diagram Alir Proses Menjalankan *Mouse*

3.4.8 Diagram Alir Proses Pembacaan *MAC Address* pada Android

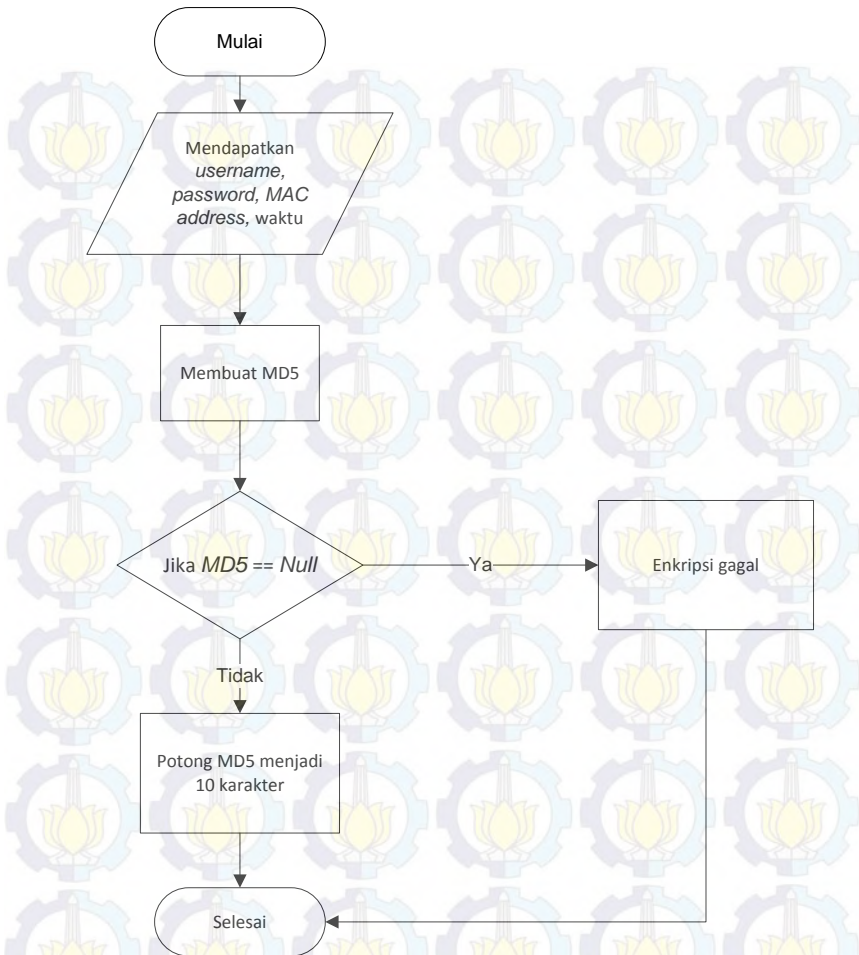
Pada Gambar 3.10 disajikan diagram alir proses pembacaan *MAC address* saat membuka aplikasi *Client* pada Android. Aplikasi langsung mendeteksi *MAC address* dari perangkat tersebut, jika Wifi tidak aktif akan diberitahukan bahwa tidak ada perangkat Wifi atau Wifi tidak aktif.



Gambar 3.10 Diagram Alir Proses Pembacaan *MAC Address* pada Android

3.4.9 Diagram Alir Proses Pembuatan Sesi MD5

Pada Gambar 3.11 disajikan diagram alir proses pembuatan sesi MD5. Proses berlangsung ketika pengguna memasukkan *username*, *password*, *MAC address*, dan *port* kemudian menyimpan pengaturan *account* tersebut, saat *server* dijalankan (*run*) maka saat itu *server* akan membuat *session code* dengan dasar *username*, *password*, *MAC address*, dan waktu.



Gambar 3.11 Diagram Alir Proses Pembuatan Sesi MD5

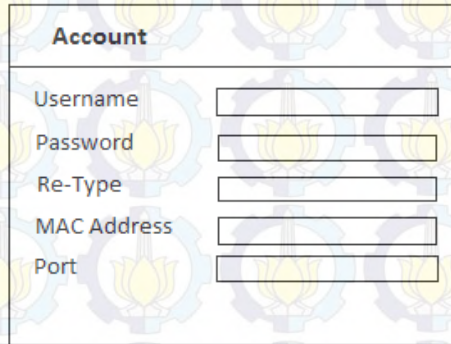
3.5 Perancangan Antarmuka

Pada tahap ini akan dipaparkan rancangan atau desain antarmuka perangkat lunak yang akan dibangun dengan

mempunyai beberapa tampilan menu yaitu menu *account*, menu *setting*, menu *log*, dan menu *control* pada aplikasi *Server* dan menu *connect*, menu *keyboard*, dan menu *mouse* pada aplikasi *Client*.

3.5.1 Rancangan Antarmuka *Account* pada Aplikasi *Server*

Rancangan antarmuka menu utama seperti pada Gambar 3.12 ini digunakan *user* untuk menghubungkan *server* dengan perangkat Android. *User* harus memasukan *username*, *password*, *MAC address*, dan *port*.

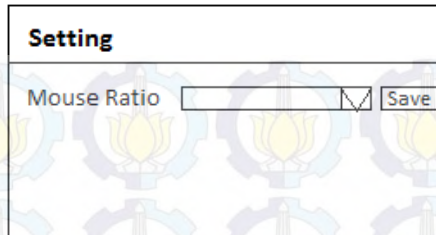


Account	
Username	<input type="text"/>
Password	<input type="password"/>
Re-Type	<input type="password"/>
MAC Address	<input type="text"/>
Port	<input type="text"/>

Gambar 3.12 Antarmuka *Account* pada *Server*

3.5.2 Rancangan Antarmuka *Setting* pada Aplikasi *Server*

Rancangan antarmuka seperti pada Gambar 3.13 ini digunakan untuk memilih *mouse ratio* yang diinginkan oleh *user*.

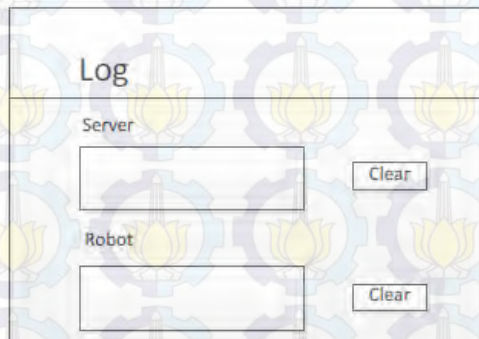


Setting	
Mouse Ratio	<input type="text"/> <input checked="" type="checkbox"/> <input type="button" value="Save"/>

Gambar 3.13 Antarmuka *Setting* pada *Server*

3.5.3 Rancangan Antarmuka *Log* pada Aplikasi *Server*

Rancangan antarmuka seperti pada Gambar 3.14 ini digunakan untuk menampilkan pergerakan *mouse* dan *keyboard* yang masuk pada *server* dan *robot*. *User* juga bisa menghapus *log* yang masuk.



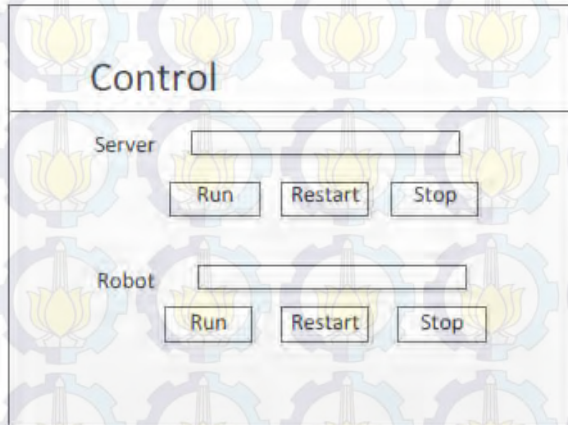
Log	
Server	<input type="text"/> <input type="button" value="Clear"/>
Robot	<input type="text"/> <input type="button" value="Clear"/>

Gambar 3.14 Antarmuka *Log* pada *Server*

3.5.4 Rancangan Antarmuka *Control* pada Aplikasi *Server*

Rancangan antarmuka seperti pada Gambar 3.15 ini berfungsi untuk menjalankan *server* dan *robot*, aplikasi baru bisa

berjalan apabila kedua fungsi tersebut telah dijalankan. *User* juga bisa menghentikan maupun memulai dari awal fungsi tersebut.



Gambar 3.15 Antarmuka *Control* pada *Server*

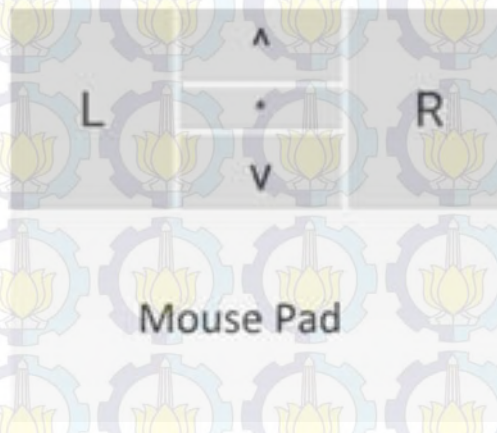
3.5.5 Rancangan Antarmuka Aplikasi *Client*

Rancangan antarmuka seperti pada Gambar 3.16 ini berfungsi untuk menampilkan menu utama pada aplikasi Android. *User* mengisi *username*, *password*, *server IP*, *server port* untuk dihubungkan dengan aplikasi di *server*.

Gambar 3.16 Antarmuka *Connect* pada *Client*

3.5.6 Rancangan Antarmuka *Mouse* pada Aplikasi *Client*

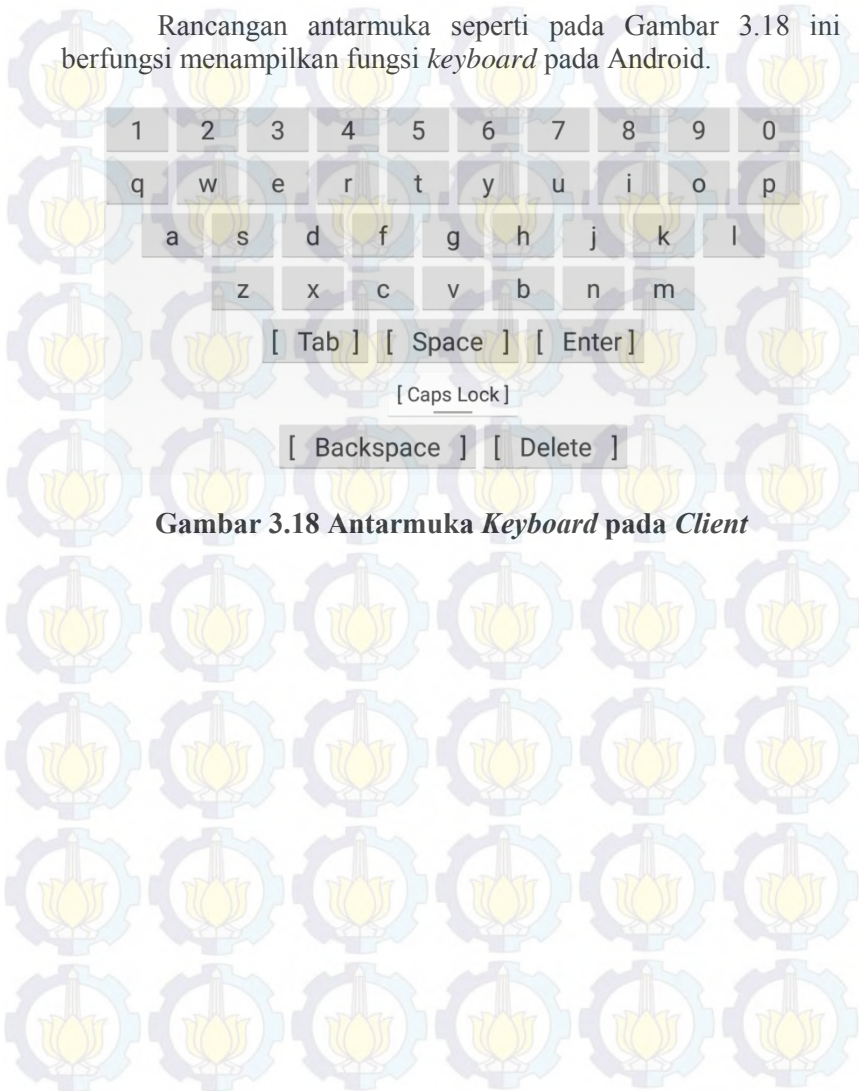
Rancangan antarmuka seperti pada Gambar 3.17 ini berfungsi menampilkan fungsi *mouse* pada Android, terdapat pilihan *scroll* dalam aplikasi ini.



Gambar 3.17 Antarmuka *Mouse* pada *Client*

3.5.7 Rancangan Antarmuka *Keyboard* pada Aplikasi *Client*

Rancangan antarmuka seperti pada Gambar 3.18 ini berfungsi menampilkan fungsi *keyboard* pada Android.



Gambar 3.18 Antarmuka *Keyboard* pada *Client*

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Pembangunan perangkat lunak dilakukan pada lingkungan pengembangan sebagai berikut:

4.1.1 Lingkungan Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk pengembangan sistem adalah sebagai berikut:

- Sistem operasi berupa Windows 7 Ultimate 64 bit
- Android Lollipop 5.0.2 sebagai sistem operasi ponsel.
- Star UML sebagai alat untuk mendesain diagram kasus.
- Eclipse Luna Service Release 2 sebagai alat untuk melakukan pemrograman.
- Bahasa pemrograman Java sebagai bahasa pemrograman untuk menulis program.

4.1.2 Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk *server* adalah laptop Dell Inspiron 14-3421, Intel Core i3-3217U 1.80 GHz dan 2 GB memory DDR3 sedangkan untuk *client* adalah *smartphone* Sony Xperia Z Ultra.

4.2 Implementasi Pembuatan Kelas

Implementasi kelas mengarah pada kelas-kelas yang telah diidentifikasi yang didasarkan dari hasil tahap analisis dan desain

sebelumnya. Pada implementasi perangkat lunak ini terdapat tujuh kelas seperti pada Tabel 4.1. Kelas dibagi menjadi 2. Pertama merupakan kelas pada *server*, kedua merupakan kelas pada *client*.

Tabel 4.1 Daftar Kelas pada Aplikasi *Server*

No	Nama Kelas pada Server	Keterangan
1	RemoteGui.java	Kelas ini untuk menampilkan button tab dan lain-lain. Fungsinya hanya untuk tampilan saja.
2	RemoteRFC.java	Kelas ini berisi tentang standar kamus untuk komunikasi dengan client.
3	RemoteRobot.java	Kelas ini berisi kode kode untuk menggerakkan fungsi mouse dan keyboard.
4	RemoteServer.java	Kelas ini berfungsi untuk melakukan komunikasi dan validasi dengan client.

Tabel 4.2 Daftar Kelas pada Aplikasi *Client*

No	Nama Kelas pada Client (Android)	Keterangan
1	RemoteActivity.java	Kelas ini digunakan untuk melakukan load tampilan Android dan memproses input dari user.
2	Remote.java	Super kelas untuk menghandle masalah UI.
3	RemoteRFC.java	Kelas ini berisi tentang standar kamus untuk komunikasi dengan server.
4	RemoteConnector.java	Kelas ini berfungsi untuk melakukan komunikasi dengan server.
5	MultiTouchActivity.java	Kelas ini berfungsi untuk menangani multi-touch pada suatu activity tertentu.

Salah satu nama kelas diatas adalah kelas RemoteRobot.java kelas ini sebuah kelas yang mengimplementasikan *thread* dan memanggil fungsi-fungsi pada kelas java.awt.robot. Tahap implemementasi kelas menggunakan bahasa pemrograman Java pada lingkungan pemrograman J2SE untuk *server* dan Java untuk Android.

4.3 Implementasi Proses Pengisian *Username, Password, MAC Address, Port* pada Aplikasi *Server*

Proses pengisian *username, password, MAC address*, dan *port* dilakukan pada awal aplikasi dibuka di komputer. Pada bagian ini *username, password, MAC address*, dan *port* akan

dimasukan dan disimpan dalam suatu direktori lalu disesuaikan dengan aplikasi yang ada pada Android. Apabila *input* tidak sesuai maka kedua perangkat tidak dapat terhubung. Source Code 4.1 merupakan *code* untuk proses menginputkan *username*, *password*, *MAC address*, dan *port* pada aplikasi *server*.

```
private void _saveActionPerformed(java.awt.event.ActionEvent evt)
{
    String the_pass=new String(_password.getPassword());
    String second_pass=new String(_repassword.getPassword());
    String the_mac=_mac_address.getText();
    String the_port=_port.getText();

    if(!the_pass.equals(second_pass)){
        JOptionPane.showMessageDialog(this,"Password Tidak
        Cocok");
        return;
    }
    if(the_mac.equals("") || the_mac.length()!=17){
        JOptionPane.showMessageDialog(this,"Mac Address
        Salah");
        return;
    }
    if(the_port.equals("")){
        JOptionPane.showMessageDialog(this,"Port Salah");
        return;
    }
    password=the_pass;
    username=_username.getText();
    mac=the_mac;
    port=Integer.parseInt(the_port);

    JOptionPane.showMessageDialog(this,"Account dan Settings
    Server Disimpan, Silakan Restart Server");
}
```

Source Code 4.1 Implementasi Pengisian *Username*, *Password*, *MAC Address*, *Port* pada *Server*

Source Code 4.1 merupakan sebuah proses pengisian *username*, *password*, *MAC address*, dan *port*. Setelah menu *account* dipilih maka akan ditampilkan sebuah *form* yang berguna untuk *input* dalam *textfield* *username*, *password*, *MAC address*, dan *port* seperti pada Source Code 4.1.

4.4 Implementasi Proses Pengaturan *Mouse Ratio*

Pengaturan *mouse ratio* yang diinginkan dilakukan dengan memilih *combo box* yang ada pada menu *setting*, pilihan tersebut lalu disimpan. Source Code 4.2 merupakan *code* untuk pengaturan *mouse ratio*.

```
private void _mouse_saveActionPerformed(java.awt.event.ActionEvent
evt) {
    String
    chooser=_mouse_chooser.getSelectedItem().toString();
    mouse_ratio=Float.parseFloat(chooser);
    JOptionPane.showMessageDialog(this, "Ratio Change to
    "+chooser+", Restart The Robot");
}
```

Source Code 4.2 Implementasi Pengaturan *Mouse Ratio*

4.5 Implementasi Proses Penambahan Daftar *Log*

Proses penambahan *log* otomatis terjadi apabila *user* menggunakan *mouse* atau *keyboard* melalui aplikasi ini. Proses tersebut disimpan ke dalam suatu direktori. *Log* yang tertera ada dua, *log* pada *server* dan *log* pada *robot*, *log* pada *server* merupakan *input* yang diterima *server* ketika *robot* dijalankan. *Log* pada *robot* merupakan alat penggerak ketika *user* menggunakan aplikasi dari Android. *User* bisa menghapus *log* dengan memilih *clear*. Lebih jelasnya tentang implementasi proses penambahan *log* dapat dilihat pada Source Code 4.3 yang merupakan *code* penambahan *log*.

```
public static final void appendLogServer(String append){
    if(LOG_SERVER!=null){
        synchronized (LOG_SERVER) {
            LOG_SERVER.append(append+"\n");
        }
    }
}

public static final void appendLogRobot(String append){
    if(LOG_ROBOT!=null){
        synchronized (LOG_ROBOT) {
```



```
LOG_ROBOT.append(append+"\n");
```

```
    }  
}
```

Source Code 4.3 Implementasi Penambahan Log

4.6 Implementasi Proses Menjalankan *Server* dan *Robot*

Proses menjalankan *server* dan *robot* dilakukan setelah *user* mengisi *textfield* yang ada pada menu *account*, lalu *user* memilih menu *control* dan menjalankan *server* yang merupakan aplikasi pada komputer dan *robot* sebagai aplikasi penggeraknya. Source Code 4.4 merupakan Implementasi *code* untuk proses menjalankan *server* dan *robot*.

```
private void _server_runActionPerformed(java.awt.event.ActionEvent  
evt) {  
    RemoteServer.startServer();  
    _server_status.setText("Server Berjalan");  
}  
  
private void  
_server_stopActionPerformed(java.awt.event.ActionEvent evt) {  
    RemoteServer.stopServer();  
    _server_status.setText("Server Berhenti");  
}  
  
private void  
_server_restartActionPerformed(java.awt.event.ActionEvent evt) {  
    _server_stopActionPerformed(evt);  
    _server_runActionPerformed(evt);  
}  
  
private void  
_robot_runActionPerformed(java.awt.event.ActionEvent evt) {  
    RemoteRobot.startRobot();  
    _robot_status.setText("Robot Berjalan");  
}  
  
private void  
_robot_stopActionPerformed(java.awt.event.ActionEvent evt) {  
    RemoteRobot.stopRobot();  
    _robot_status.setText("Robot Berhenti");  
}
```

```

    private void
    _robot_restartActionPerformed(java.awt.event.ActionEvent evt) {
        _robot_stopActionPerformed(evt);
        _robot_runActionPerformed(evt);
    }

```

Source Code 4.4 Implementasi Proses Menjalankan *Server* dan *Robot*

4.7 Implementasi Proses Pengisian *Username*, *Password*, *Server IP*, *Server Port* pada Android

Proses pengisian *username*, *password*, *server IP*, dan *server port* dilakukan pada awal aplikasi dibuka di Android. Pada bagian ini *username*, *password*, MAC address, *server IP* dan *port* akan disimpan dalam suatu direktori lalu disesuaikan dengan aplikasi yang ada pada komputer. Apabila *input* tidak sesuai maka kedua perangkat tidak dapat terhubung. Lebih jelasnya tentang implementasi fungsi pengisian *username*, *password*, *server IP*, dan *server port* dijelaskan pada Source Code 4.5. Source Code 4.5 merupakan Implementasi *code* untuk memasukan *username*, *password*, *server IP*, dan *server port* serta pengecekan *login* pada Android.

```

    private class LoginTask extends AsyncTask<Void, String,
String> {
        ProgressDialog progress;
        protected void onPreExecute() {
            TextView username=(TextView)
            findViewById(R.id.username);
            TextView password=(TextView)
            findViewById(R.id.password);
            TextView port=(TextView)
            findViewById(R.id.port);
            TextView ip=(TextView)
            findViewById(R.id.ip);

            _username=username.getText().toString();
            _password=password.getText().toString();
            _ip=ip.getText().toString();

```

```

        _port=port.getText().toString();

        RemoteConnector.setURL(_ip, _port);
        progress=
ProgressDialog.show(RemoteActivity.this,
        "",getString(R.string.Loading));
        progress.setCancelable(true);
        progress.setOnCancelListener(new
OnCancelListener() {
            @Override
            public void
onCancel(DialogInterface dialog) {
                cancel(true);
            }
        });
        @Override
        protected void onProgressUpdate(String... values)
        {
        }
        @Override
        protected String doInBackground(Void ...param) {
            publishProgress("Sending Request");
            return
RemoteConnector.Login(_username+"."+_password+"."+_mac_address);
        };
        @Override
        protected void onPostExecute(String result) {
            progress.dismiss();
            String show="";
            if(!result.equals("FAIL")){
                show=getString(R.string.remote_Login_succes
s);
            }else{
                show=getString(R.string.remote_Login_fail);
            }
            Toast.makeText(getApplicationContext(),
show, Toast.LENGTH_SHORT).show();
        }
    };

```

Source Code 4.5 Implementasi Proses Pengisian Username, Password, Server IP, Server Port pada Android

4.8 Implementasi Proses Menjalankan *Keyboard*

Proses menjalankan *keyboard* bisa terlaksana apabila *login* antara *server* dengan perangkat Android telah berhasil, lalu *user* memilih *keyboard* yang ada di aplikasi Android setelah itu akan keluar tampilan tombol *keyboard* yang dapat dijalankan. Source Code 4.6 merupakan Implementasi *code* untuk *keyboard* di Android. Source Code 4.7 merupakan Implementasi *code* untuk *keyboard* pada *server*.

```
private boolean keyboardKey(View v){
    if(!isConnected()) return false;
    Button b=(Button) v;
    String text=b.getText().toString();
    String
before=this.key_control?RemoteRFC.CONTROL:"";
before+=this.key_alt?RemoteRFC.ALT:"";
before+=this.key_shift?RemoteRFC.SHIFT:"";
    return
RemoteConnector.sendCommand(before+RemoteRFC.KEYBOARD+text);
}

private void capsLock(View v){
    ToggleButton tb=(ToggleButton)
findViewById(R.id.caps_key);
    boolean check=tb.isChecked();
    if(check){
        LinearLayout fk=(LinearLayout)
findViewById(R.id.first_key);
        for(int i=0;i<fk.getChildCount();i++){
            Button b=(Button)
fk.getChildAt(i);
            String
text=b.getText().toString().toUpperCase();
            b.setText(text);
        }
        LinearLayout sk=(LinearLayout)
findViewById(R.id.second_key);
        for(int i=0;i<sk.getChildCount();i++){
            Button b=(Button)
sk.getChildAt(i);
            String
text=b.getText().toString().toUpperCase();
            b.setText(text);
        }
    }
}
```



```

        LinearLayout tk=(LinearLayout)
        findViewById(R.id.third_key);
        for(int i=0;i<tk.getChildCount();i++){
            Button b=(Button)
            tk.getChildAt(i);
            String
            text=b.getText().toString().toUpperCase();
            b.setText(text);
        }
    }else{
        LinearLayout fk=(LinearLayout)
        findViewById(R.id.first_key);
        for(int i=0;i<fk.getChildCount();i++){
            Button b=(Button)
            fk.getChildAt(i);
            String
            text=b.getText().toString().toLowerCase();
            b.setText(text);
        }
        LinearLayout sk=(LinearLayout)
        findViewById(R.id.second_key);
        for(int i=0;i<sk.getChildCount();i++){
            Button b=(Button)
            sk.getChildAt(i);
            String
            text=b.getText().toString().toLowerCase();
            b.setText(text);
        }
        LinearLayout tk=(LinearLayout)
        findViewById(R.id.third_key);
        for(int i=0;i<tk.getChildCount();i++){
            Button b=(Button)
            tk.getChildAt(i);
            String
            text=b.getText().toString().toLowerCase();
            b.setText(text);
        }
    }
}

private boolean specialKey(View v){
    if(!isConnected()) return false;

    Button b=(Button) v;
    String text=b.getText().toString();
    int data=0;
    if(text.contains("Enter"))
    text.contains("ENTER") || text.contains("enter")){
        ||
    }
}

```

```

        data=10;
    }else if(text.contains("Space")){
        data=32;//
    }else if(text.contains("Tab") ||
text.contains("TAB") || text.contains("tab")){
        data=9;
    }else if(text.contains("Backspace")){
        data=8;
    }else if(text.contains("Delete") ||
text.contains("DELETE") || text.contains("delete")){
        data=127;
    }else{
        return false;
    }
    String
before=this.key_control?RemoteRFC.CONTROL:"";
before+=this.key_alt?RemoteRFC.ALT:"";
before+=this.key_shift?RemoteRFC.SHIFT:"";
    return
RemoteConnector.sendCommand(before+RemoteRFC.SPECIAL+data);
}

```

Source Code 4.6 Implementasi Proses Menjalankan *Keyboard* pada Android

```

public static void type(int i){
    ROBOT.delay(40);
    if(RemoteRobot._control)
ROBOT.keyPress(KeyEvent.VK_CONTROL);
    if(RemoteRobot._shift)
ROBOT.keyPress(KeyEvent.VK_SHIFT);
    if(RemoteRobot._alt)
ROBOT.keyPress(KeyEvent.VK_ALT);
    ROBOT.keyPress(i);
    ROBOT.keyRelease(i);
    if(RemoteRobot._control)
ROBOT.keyRelease(KeyEvent.VK_CONTROL);
    if(RemoteRobot._shift)
ROBOT.keyRelease(KeyEvent.VK_SHIFT);
    if(RemoteRobot._alt)
ROBOT.keyRelease(KeyEvent.VK_ALT);
    RemoteRobot._control=false;
    RemoteRobot._shift=false;
    RemoteRobot._alt=false;
}

```

```

    }

    public static void type(String s){
        byte[] bytes = s.getBytes();
        for (byte b : bytes){
            int code = b;
            ROBOT.delay(40);
            if(RemoteRobot._control)
                ROBOT.keyPress(KeyEvent.VK_CONTROL);
            if(RemoteRobot._shift)
                ROBOT.keyPress(KeyEvent.VK_SHIFT);
            if(RemoteRobot._alt)
                ROBOT.keyPress(KeyEvent.VK_ALT);
            ROBOT.keyPress(code);
            ROBOT.keyRelease(code);
            if(RemoteRobot._control)
                ROBOT.keyRelease(KeyEvent.VK_CONTROL);
            if(RemoteRobot._shift)
                ROBOT.keyRelease(KeyEvent.VK_SHIFT);
            if(RemoteRobot._alt)
                ROBOT.keyRelease(KeyEvent.VK_ALT);
            RemoteRobot._control=false;
            RemoteRobot._shift=false;
            RemoteRobot._alt=false;
        }
    }

    public static void typeCharacter(Robot robot, String
    letter){
        try{
            boolean upperCase =
            Character.isUpperCase(letter.charAt(0));
            String variableName = "VK_" + letter.toUpperCase();
            Class clazz = KeyEvent.class;
            Field field = clazz.getField( variableName );
            int keyCode = field.getInt(null);
            robot.delay(40);
            if (upperCase) robot.keyPress( KeyEvent.VK_SHIFT );

            if(RemoteRobot._control)
                ROBOT.keyPress(KeyEvent.VK_CONTROL);
            if(RemoteRobot._shift)
                ROBOT.keyPress(KeyEvent.VK_SHIFT);
            if(RemoteRobot._alt)
                ROBOT.keyPress(KeyEvent.VK_ALT);

```



```

        robot.keyPress( keyCode );
        robot.keyRelease( keyCode );
        if (upperCase) robot.keyRelease( KeyEvent.VK_SHIFT );
        if(RemoteRobot._control)
        ROBOT.keyRelease(KeyEvent.VK_CONTROL);
        if(RemoteRobot._shift)
        ROBOT.keyRelease(KeyEvent.VK_SHIFT);
        if(RemoteRobot._alt)
        ROBOT.keyRelease(KeyEvent.VK_ALT);
        RemoteRobot._control=false;
        RemoteRobot._shift=false;
        RemoteRobot._alt=false;
    }catch(Exception e){}
}

```

Source Code 4.7 Implementasi Proses *Keyboard* pada *Server*

4.9 Implementasi proses Multi-touch pada *Keyboard*

Source code ini menangkap sentuhan jari user. Kemudian mendeteksi apakah ACTION MOVE (menyentuh layar dan menggerakkan jari) atau bukan. Proses multi-touch dilakukan ketika satu jari menekan dan menahan tombol *keyboard* kemudian jari yang lain menekan tombol lain pada *keyboard*. Source Code 4.8 merupakan Implementasi proses multi-touch pada *keyboard*.

```

public boolean onTouch(final View v, final MotionEvent event) {
    // index of the pointer which starts this Event
    final int actionPointerIndex =
event.getActionIndex();

    // resolve the action as a basic type (up, down or
move)
    int actionResolved = event.getAction() &
MotionEvent.ACTION_MASK;
    if (actionResolved < 7 && actionResolved > 4) {
        actionResolved = actionResolved - 5;
    }
    if (actionResolved == MotionEvent.ACTION_MOVE) {
        for (int ptrIndex = 0; ptrIndex <
event.getPointerCount(); ptrIndex++) {

            // only one event for all move events.

```



```

dealEvent(ptrIndex, event, v,
actionResolved);
    }
} else {
    dealEvent(actionPointerIndex, event, v,
actionResolved);
}
return true;
}

```

Source Code 4.8 Implementasi Proses Multi-touch pada Keyboard

4.10 Implementasi Proses Menjalankan Mouse

Proses menjalankan *mouse* dilakukan setelah *login* koneksi antara *server* dengan perangkat Android telah berhasil terhubung, kemudian *user* memilih menu *mouse* yang ada di aplikasi Android. Setelah itu akan keluar tampilan tombol pada *mouse* dan *touchpad mouse* yang dapat dijalankan. Source Code 4.9 merupakan Implementasi *mouse gesture* pada *server*. Source Code 4.10 merupakan Implementasi *mouse client* pada Android. Source Code 4.11 merupakan Implementasi tombol *mouse* pada *server*.

```

the_order=the_order.substring(3);
String[] xy=the_order.split(",");
int dx=Integer.parseInt(xy[0])*(-1);
int dy=Integer.parseInt(xy[1])*(-1);
if(dx==0 && dy==0){
    LeftClick();
    continue;
}
PointerInfo pointer=MouseInfo.getPointerInfo();
Point p=pointer.getLocation();
int x1=(int)p.getX();
int y1=(int)p.getY();
int x2=x1+(int)(dx*ratio);
int y2=y1+(int)(dy*ratio);
ROBOT.mouseMove(x2, y2);

```

Source Code 4.9 Implementasi Mouse Gesture pada Server

```

public boolean onTouch(View view, MotionEvent event) {
    int id=view.getId();
    if(id==R.id.control){

        if(event.getAction()==MotionEvent.ACTION_DOWN){
            this.key_control=true;

        }else
        if(event.getAction()==MotionEvent.ACTION_UP ||
        event.getAction()==MotionEvent.ACTION_CANCEL){
            this.key_control=false;
        }
        if(event.getActionIndex()==0) return
        false;

        return super.onTouch(view, event);
    }else if(id==R.id.alt){

        if(event.getAction()==MotionEvent.ACTION_DOWN){
            this.key_alt=true;

        }else
        if(event.getAction()==MotionEvent.ACTION_UP ||
        event.getAction()==MotionEvent.ACTION_CANCEL){
            this.key_alt=false;
        }
        if(event.getActionIndex()==0) return
        false;

        return super.onTouch(view, event);
    }else if(id==R.id.shift_button){

        if(event.getAction()==MotionEvent.ACTION_DOWN){
            this.key_shift=true;

        }else
        if(event.getAction()==MotionEvent.ACTION_UP ||
        event.getAction()==MotionEvent.ACTION_CANCEL){
            this.key_shift=false;
        }
        if(event.getActionIndex()==0) return
        false;

        return super.onTouch(view, event);
    }else{
        if(!isConnected()) return false;
        switch(event.getAction()){
            case MotionEvent.ACTION_DOWN : x1=event.getX();
                                            y1=event.getY();
                                            break;

            case MotionEvent.ACTION_UP :
            case MotionEvent.ACTION_MOVE : x2=event.getX();

```

```

        y2=event.getY();
        int dx=(int)(x1-
        int dy=(int)(y1-
        String
        command=RemoteRFC.GESTURE+dx+" "+dy;
        RemoteConnector.sendCommand(command);
        break;
    }
    return true;
}

```

Source Code 4.10 Implementasi Proses *Mouse* pada *Client*

```

public static void leftClick(){
    ROBOT.mousePress(InputEvent.BUTTON1_MASK);
    ROBOT.delay(200);
    ROBOT.mouseRelease(InputEvent.BUTTON1_MASK);
    ROBOT.delay(200);
}
public static void middleClick(){
    ROBOT.mousePress(InputEvent.BUTTON2_MASK);
    ROBOT.delay(200);
    ROBOT.mouseRelease(InputEvent.BUTTON2_MASK);
    ROBOT.delay(200);
}
public static void rightClick(){
    ROBOT.mousePress(InputEvent.BUTTON3_MASK);
    ROBOT.delay(200);
    ROBOT.mouseRelease(InputEvent.BUTTON3_MASK);
    ROBOT.delay(200);
}
public static void mouseWheel(int up){
    ROBOT.mouseWheel(up);
}

```

Source Code 4.11 Implementasi Tombol *Mouse* pada *Server*

4.11 Implementasi Proses Pembacaan *MAC Address* pada Android

Proses pengambilan *MAC address* oleh aplikasi *client* pada android dilakukan pada saat perangkat terhubung dengan Wifi dan pengguna membuka aplikasi tersebut. Aplikasi secara otomatis mendeteksi *MAC address* dari perangkat tersebut. Source Code 4.12 merupakan Implementasi pembacaan *MAC address* pada perangkat Android.

```
public String getMacAddress(Context context) {
    WifiManager wimanager = (WifiManager)
context.getSystemService(Context.WIFI_SERVICE);
    String macAddress =
wimanager.getConnectionInfo().getMacAddress();
    if (macAddress == null) {
        macAddress = getString(R.string.remote_device);
    }
    return macAddress;
}
```

Source Code 4.12 Implementasi Pembacaan *MAC Address* pada Android

4.12 Implementasi Proses Pembuatan Sesi MD5

Proses pembuatan sesi MD5 dilakukan oleh *server* ketika *server* mendapatkan *username*, *password*, *MAC address* dari menu *account* pada aplikasi *server*. Kemudian ketika *server* dijalankan, proses MD5 dilakukan pada *username*, *password*, *MAC address*, dan waktu dibutuhkan untuk membuat *session code*. Sehingga ketika jaringan terputus atau *server restart*, *session code* berubah karena terdapat waktu di sana. Source Code 4.13 merupakan Implementasi proses pembuatan sesi MD5.

```
public static String MD5(String md5) {
    try {
        java.security.MessageDigest md =
java.security.MessageDigest.getInstance("MD5");
        byte[] array = md.digest(md5.getBytes());
```



```

        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < array.length; ++i) {
            sb.append(Integer.toHexString((array[i]
& 0xFF) | 0x100).substring(1,3));
        }
        return sb.toString().substring(0, 10);
    } catch
    (java.security.NoSuchAlgorithmException e) {
    }
    return "FAIL";
}

```

Source Code 4.13 Implementasi Pembuatan Sesi MD5

4.13 Implementasi Proses Pembuatan Kelas RemoteRobot.java

Kelas RemoteRobot.java adalah sebuah kelas yang mengimplementasikan *thread* dan memanggil fungsi pada kelas java.awt.robot. Kelas ini adalah sebuah *thread* yang diatur sebagai *daemon* yang berjalan secara terus-menerus untuk membaca isi dari *mutex*. setiap kali pembacaan *mutex*, *server* akan mendapatkan sebuah pesan (*command*) yang telah dikumpulkan oleh RemoteServer, kelas *robot* akan menerjemahkan isi dari pesan tersebut dalam bentuk gerakan-gerakan *mouse* dan *keyboard* dengan memanfaatkan kelas java.awt.robot.

```

public void run(){
    while(alive){
        try{
            String the_order=null;
            synchronized (RemoteGui.MUTEX) {
                the_order=RemoteGui.MUTEX.pollLast();
            }
            if(the_order!=null){
                RemoteGui.appendLogRobot(the_order);
                if(the_order.contains(RemoteRFC.MOUSE)){
                    if(the_order.contains(RemoteRFC.CLICK_LEFT)){
                        leftClick();
                    }else if(the_order.contains(RemoteRFC.CLICK_RIGHT)){
                        rightClick();
                    }else if(the_order.contains(RemoteRFC.CLICK_MIDDLE)){
                        middleClick();
                    }
                }
            }
        }
    }
}

```

```
}else if(the_order.contains(RemoterFC.SCROLL_DOWN)){  
    mousewheel(1);  
}else if(the_order.contains(RemoterFC.SCROLL_UP)){  
    mousewheel(-1);  
}  
}
```

Source Code 4.14 Potongan *Code* Kelas RemoteRobot.java

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kebenaran dan uji kinerja serta analisa setiap pengujian.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan aplikasi. Lingkungan Uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Laptop Dell Inspiron 14-3421
Prosesor: Intel® Core™ i3-3217U CPU @ 1.80 GHz
RAM 2 GB
 - b. Ponsel Android Sony Xperia Z Ultra
Prosesor: Qualcomm MSM8274 Quad-core 2.2 GHz
Krait 400 RAM 2 GB
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Ultimate 64 bit untuk *server* dan Android Lollipop 5.0.2 untuk *client*
 - b. Perangkat pengembang:
 - Java SE Development Kit 8 Update 31
 - Eclipse Luna Service Release 2.
 - Android SDK Revision 24.3.2

5.2 Skenario Uji Coba

Uji coba aplikasi Tugas Akhir ini terdiri dari uji coba fungsionalitas program. Uji coba fungsionalitas program dilakukan untuk menguji fungsi dari tiap kontrol untuk setiap fitur

yang ada pada aplikasi sesuai dengan diagram kasus yang terdapat pada tahap perancangan perangkat lunak.

5.2.1 Uji Coba Fungsionalitas

Uji coba fungsionalitas program dilakukan untuk menguji fungsi dari tiap kontrol untuk setiap fitur yang ada pada aplikasi. Pada tugas akhir ini, uji coba aplikasi dilakukan pada *server* dan *client*. Pada *server*, uji coba terdiri dari menu *account* untuk pengisian *username*, *password*, *MAC address*, *server port*, menu *setting* untuk pengaturan *mouse ratio*, menu *log* untuk melihat dan menghapus *log*, menu *control* untuk menjalankan *server* dan *robot*. Sedangkan pada aplikasi *client*, uji coba terdiri dari menu *connect* untuk pengisian *username*, *password*, *server IP*, dan *server port* pada Android serta menu *keyboard* untuk menjalankan *keyboard* dan menu *mouse* untuk menjalankan *mouse*.

5.2.1.1 Uji Coba Pengisian Username, Password, MAC Address, dan Port

Uji coba ini dilakukan dengan memasukkan *username*, *password*, *MAC address*, dan *port* pada menu *account* seperti Gambar 5.1 yang nantinya akan disimpan dan disamakan dengan *username*, *password*, *MAC address*, *port* yang ada di Android. Setelah *textfield* diisi, *save* dipilih untuk menyimpan data tersebut. Rangkuman dari proses uji coba pengisian *username*, *password*, *MAC address*, *server port* dapat dilihat pada Tabel 5.1. Contoh pengisian *textfield* bisa dilihat pada Gambar 5.1.

Tabel 5.1 Uji Coba Pengisian *Username, Password, MAC Address, Port*

Kode Diagram Kasus	UJI-01
Nama uji coba	Uji coba pengisian username, password, MAC address, dan port.
Tujuan uji coba	Menguji fitur pengisian username, password, MAC address, dan port.
Skenario	Pengguna mengisi username, password, MAC address, dan port.
Kondisi awal	Perangkat berada pada menu account aplikasi Server.
Kondisi akhir	Perangkat berada pada menu account aplikasi Server.
Hasil uji coba	Berhasil.

The screenshot shows a window titled "(Server Remote Control)". It has four tabs: "Account", "Settings", "Log", and "Control". The "Account" tab is active. The form contains the following fields:

- Username : saiful
- Password : *****
- Re-Type : *****
- Mac Address : 4c:21:d0:49:9b:28
- Port : 8888

A "Save" button is located at the bottom right of the form.

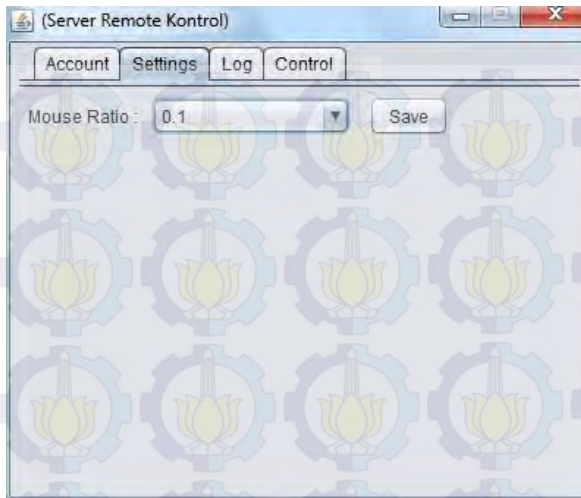
Gambar 5.1 Antarmuka Pemasukan *Username, Password, MAC Address, dan Server Port* pada *Server*

5.2.1.2 Uji Coba Pengaturan *Mouse Ratio*

Uji coba ini dilakukan dengan memilih menu *setting* dan memilih *mouse ratio* yang ada pada *combo box*. Setelah dipilih maka pilihan tersebut disimpan dengan memilih *save* lalu menjalankan ulang *robot* yang ada di menu *control*. Gambar 5.2 menunjukkan pengaturan *mouse ratio* dan yang tertera di dalam *combo box* merupakan satuan *mouse ratio* yang dipilih dalam milidetik. Rangkuman dari proses uji coba dapat dilihat pada Tabel 5.2.

Tabel 5.2 Uji Coba Pengaturan *Mouse Ratio*

Kode Diagram Kasus	UJI-02
Nama uji coba	Uji coba pengaturan mouse ratio.
Tujuan uji coba	Menguji perubahan mouse ratio pada aplikasi Server.
Skenario	Pengguna mengganti mouse ratio pada aplikasi Server.
Kondisi awal	Perangkat berada pada menu setting aplikasi Server.
Kondisi akhir	Perangkat berada pada menu setting aplikasi Server.
Hasil uji coba	Berhasil.



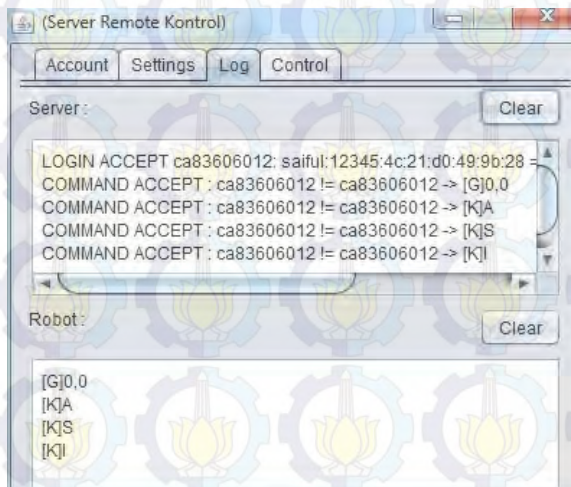
Gambar 5.2 Antarmuka Menu *Setting*

5.2.1.3 Uji Coba Penambahan Daftar *Log*

Uji coba ini dilakukan setelah aplikasi berhasil dijalankan. *Log* dapat dilihat pada menu *log* pada aplikasi komputer. Setelah *user* sukses menghubungkan perangkat Android dengan komputer dan menjalankan *keyboard* dan *mouse* maka proses tersebut akan dicatat ke dalam sebuah *log*. *Log* tersebut dibagi dua yaitu *log server* dan *log robot*. *Log server* merupakan daftar proses pada *server* komputer, *log robot* merupakan daftar proses pada *robot* sebagai aplikasi penggerak. Pengguna bisa menghapus daftar *log* pada *server* dan *robot* dengan memilih *clear*. Rangkuman dari proses uji coba penambahan dan penghapusan *log* dapat dilihat pada Tabel 5.3. Gambar 5.3 menunjukkan proses *server* dan *robot* bekerja dan tersimpan di dalam *log*.

Tabel 5.3 Uji Coba Penambahan Daftar Log

Kode Diagram Kasus	UJI-03
Nama uji coba	Uji coba penambahan dan penghapusan log.
Tujuan uji coba	Menguji fitur penambahan dan penghapusan log pada aplikasi Server.
Skenario	Pengguna menjalankan aplikasi dan proses tersebut tercatat dalam daftar log dan log tersebut dihapus.
Kondisi awal	Perangkat berada pada pilihan menu log aplikasi Server.
Kondisi akhir	Perangkat berada pada menu log aplikasi Server dengan log server dan log robot mencatat proses yang berlangsung.
Hasil uji coba	Berhasil.

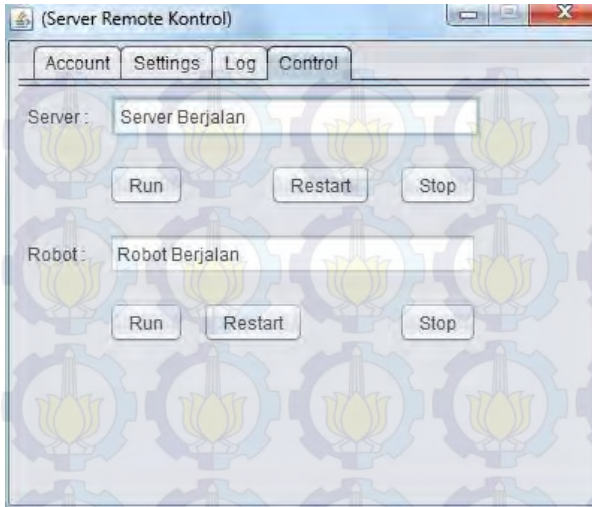
**Gambar 5.3 Antarmuka Menu Log**

5.2.1.4 Uji Coba Menjalankan *Server* dan *Robot*

Uji coba ini dilakukan dengan tujuan agar aplikasi dapat berjalan dengan benar. Setelah memasukkan *username*, *password*, *MAC address*, dan *port*, pengguna membuka menu *control*. Setelah membuka menu *control*, *server* dan *robot* dijalankan. Proses tersebut juga dapat dihentikan atau diulang dari awal. Rangkuman dari proses uji coba menjalankan *server* dan *robot* dapat dilihat pada Tabel 5.4. Gambar 5.4 menunjukkan proses *server* dan *robot* bekerja.

Tabel 5.4 Uji Coba Menjalankan *Server* dan *Robot*

Kode Diagram Kasus	UJI-04
Nama uji coba	Uji coba menjalankan server dan robot.
Tujuan uji coba	Menguji server dan robot
Skenario	Pengguna menjalankan server dan robot dan menghentikannya setelah mengisi textfield yang tersedia pada menu account.
Kondisi awal	Perangkat berada pada pilihan menu control pada aplikasi Server.
Kondisi akhir	Perangkat berada pada pilihan menu control pada aplikasi Server.
Hasil uji coba	Berhasil.



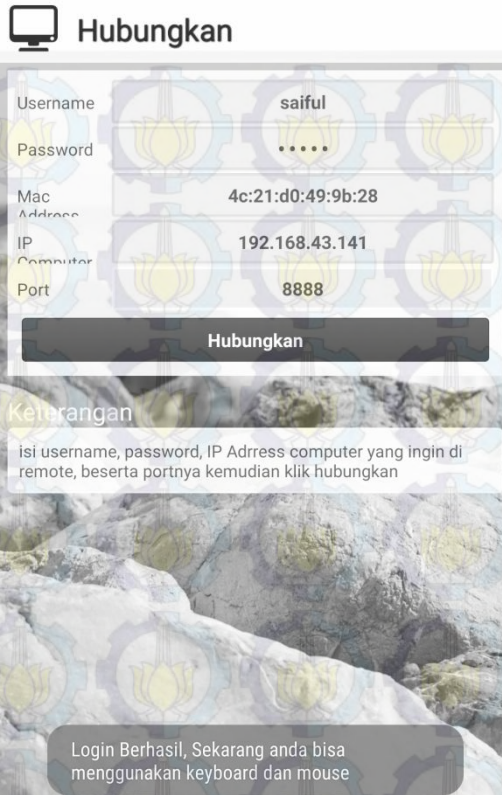
Gambar 5.4 Antarmuka Menu *Control*

5.2.1.5 Uji Coba Pengisian *Username, Password, Server IP, Server Port* pada Android

Uji coba ini dilakukan untuk menguji kesamaan antara *username, password, server IP*, dan *server port* yang ada pada Android dengan yang ada pada aplikasi komputer. Apabila berhasil maka aplikasi bisa dijalankan, apabila gagal maka aplikasi tidak dapat dijalankan. Rangkuman dari proses uji coba pengisian *username, password, server IP, server port* dapat dilihat pada Tabel 5.5. Gambar 5.5 menunjukkan proses pengisian tersebut.

Tabel 5.5 Uji Coba Pengisian *Username, Password, Server IP, Server Port* pada Android

Kode Diagram Kasus	UJI-05
Nama uji coba	Uji coba pengisian username, password, server IP, server port pada Android.
Tujuan uji coba	Menguji otentikasi aplikasi Android dengan aplikasi yang ada pada komputer.
Skenario	Pengguna mengisi username, password, server IP, server port yang sesuai dan yang berbeda dengan aplikasi komputer.
Kondisi awal	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu Connect pada perangkat Android dengan status berhasil atau gagal terhubung.
Hasil uji coba	Berhasil dan gagal.



Gambar 5.5 Antarmuka Pemasukan *Username*, *Password*, *Server IP*, *Server Port* pada Android

5.2.1.6 Uji Coba Menjalankan *Keyboard*

Uji coba ini dilakukan untuk menguji kesamaan antara *keyboard* yang ditekan pada aplikasi Android dengan hasil yang keluar pada komputer. Tampilan *keyboard* bisa dilihat seperti pada Gambar 5.6. Apabila perangkat tidak terhubung maka aplikasi tidak dapat dijalankan. Rangkuman dari proses uji coba menjalankan *keyboard* dapat dilihat pada Tabel 5.6.

Tabel 5.6 Uji Coba Menjalankan *Keyboard*

Kode Diagram Kasus	UJI-06
Nama uji coba	Uji coba menjalankan keyboard.
Tujuan uji coba	Menguji kesamaan hasil antara keyboard yang ditekan pada aplikasi Android dengan hasil yang keluar pada komputer.
Skenario	Pengguna menjalankan keyboard pada aplikasi Android setelah login berhasil.
Kondisi awal	Perangkat berada pada pilihan menu keyboard pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu keyboard pada perangkat Android.
Hasil uji coba	Berhasil.

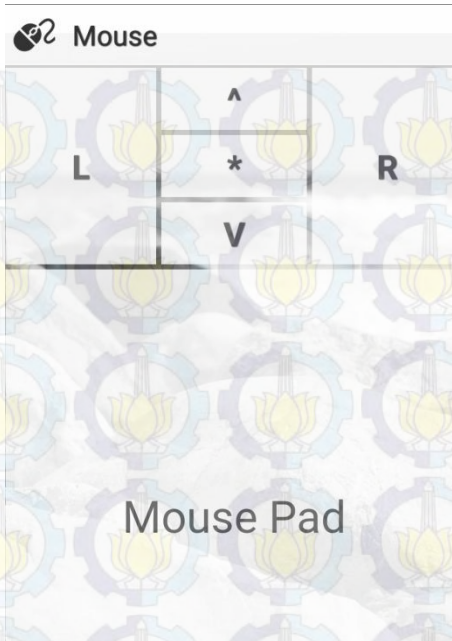
**Gambar 5.6 Antarmuka Menu *Keyboard* pada Android**

5.2.1.7 Uji Coba Menjalankan *Mouse*

Uji coba ini dilakukan untuk menguji kesamaan antara *mouse* yang dijalankan pada aplikasi Android dengan hasil yang keluar pada komputer. Tampilan *mouse* bisa dilihat seperti pada Gambar 5.7. Apabila perangkat tidak terhubung maka aplikasi tidak dapat dijalankan. Rangkuman dari proses uji coba menjalankan mouse dapat dilihat pada Tabel 5.7

Tabel 5.7 Uji Coba Menjalankan *Mouse*

Kode Diagram Kasus	UJI-07
Nama uji coba	Uji coba menjalankan mouse.
Tujuan uji coba	Menguji kesamaan hasil antara mouse yang dijalankan pada aplikasi Android dengan hasil yang keluar pada komputer.
Skenario	Pengguna menjalankan mouse pada aplikasi Android setelah login berhasil.
Kondisi awal	Perangkat berada pada pilihan menu mouse pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu mouse pada perangkat Android.
Hasil uji coba	Berhasil.



Gambar 5.7 Antarmuka Menu *Mouse* pada Android

5.2.1.8 Uji Coba Membaca *MAC Address* pada Android

Uji coba ini dilakukan untuk menguji *MAC address* yang diambil dari perangkat Android oleh aplikasi *client*. Apakah bisa mendapatkan *MAC address* dan menampilkannya atau tidak. Tampilan *MAC address* bisa dilihat seperti pada Gambar 5.8. Rangkuman dari proses uji coba membaca *MAC address* dapat dilihat pada Tabel 5.8.

Tabel 5.8 Uji Coba Membaca *MAC Address* pada Android

Kode Diagram Kasus	UJI-08
Nama uji coba	Uji coba membaca MAC address Pada Android.
Tujuan uji coba	Menguji kesamaan hasil antara MAC address pada perangkat yang sebenarnya dengan MAC address yang tampil pada aplikasi.
Skenario	Pengguna menjalankan aplikasi setelah menghidupkan Wifi.
Kondisi awal	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Hasil uji coba	Berhasil.

5.2.2 Uji Coba Kebenaran *Login, Mouse, Keyboard*

Uji coba ini dilakukan untuk menguji kebenaran *login* pada aplikasi untuk otentikasi, serta *keyboard* dan *mouse* sebagai aplikasi penggerak komputer. Pada Tugas Akhir ini, dilakukan uji coba pada otentikasi *login*, *keyboard*, dan *mouse* pada aplikasi Android.

5.2.2.1 Uji Coba pada *Login*

Pada skenario ini dipaparkan tahap pengujian perangkat lunak untuk melakukan pengecekan *login* yaitu:

1. User memasukan *username*, *password*, *server IP*, *server port* yang sesuai dengan *server*.
2. *User* memilih *connect*.
3. Perangkat terhubung, apabila ada bagian yang salah, aplikasi akan meminta pengecekan kembali data yang dimasukan.

Pada kali ini dilakukan pengujian pengecekan *login* seperti pada Tabel 5.9 dimana masukan pada *server* harus sama dengan masukan pada *Android*.

Tabel 5.9 Uji Coba pada *Login*

No	Input	Server	Client	Hasil
1	Username	Saiful	Saiful	Sukses
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
2	Username pada client tidak cocok dengan server			Gagal
	Username	Saiful	Ipunk	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
3	Password pada client tidak cocok dengan server			Gagal
	Username	Saiful	Saiful	
	Password	11111	22222	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
4	MAC Address pada server tidak cocok dengan client			Gagal
	Username	Saiful	Saiful	
	Password	11111	11111	
	MAC Address	00:11:22:C4:C4:C3	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
5	IP Server pada client tidak cocok dengan server			Gagal
	Username	Saiful	Saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.123	
	Port	8888	8888	

6	Port pada client tidak cocok dengan server			Gagal
	Username	Saiful	saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	9999	

Untuk tahapan melakukan pengecekan *login*, setelah *server* dan perangkat Android terhubung dengan Wifi, *server* akan mengecek data yang ada pada aplikasi Android apakah sama dengan data yang ada di *server*, jika sama maka koneksi tersambung, jika tidak maka aplikasi akan meminta untuk melakukan pengecekan ulang.

5.2.2.2 Uji Coba Tombol *Keyboard* pada Android

Pada skenario ini dipaparkan tahap pengujian perangkat lunak untuk melakukan pengecekan *keyboard* yaitu tombol *keyboard* dijalankan pada Android lalu dilihat hasilnya, apakah hasilnya sesuai dengan yang di komputer.

Pada tahap ini dilakukan pengujian kebenaran tombol *keyboard* yang dijalankan seperti pada Tabel 5.8, dimana masukan yang digunakan adalah tombol aplikasi *keyboard* pada Android dan hasilnya berupa tampilan pada komputer.

Tabel 5.10 Uji Coba Tombol *Keyboard* pada Android

No	Tombol Keyboard yang Ditekan	Hasil yang Keluar pada Komputer	Hasil
1	abcdefghijklmnopqrstuvwxyz	Menetik huruf abcdefghijklmnopqrstuvwxyz	Sukses
2	1234567890	Menetik angka 1234567890	Sukses
3	[Enter]	Pada layar teks kursor turun 1 baris	Sukses
4	[Tab]	Memindahkan kursor dalam 1 tabulasi ke arah kanan	Sukses
5	[Space]	Membuat jarak 1 karakter antar kata	Sukses
6	[Backspace]	Menghapus 1 karakter terakhir yang telah diketik	Sukses
7	[Delete]	File yang dipilih dihapus	Sukses
8	[Caps lock]	Sesuai dengan kondisi Caps lock	Sukses
9	[Ctrl] + a	Menjalankan perintah "Select all"	Sukses
10	[Ctrl] + c	Menjalankan perintah "Copy"	Sukses
11	[Ctrl] + v	Menjalankan perintah "Paste"	Sukses
12	[Ctrl] + x	Menjalankan perintah "Cut"	Sukses
13	[Ctrl] + z	Menjalankan perintah "Undo"	Sukses

Dapat dilihat dari Tabel 5.10 bahwa tombol *keyboard* yang ditekan pada Android sesuai dengan hasil yang keluar pada komputer.

5.2.2.3 Uji Coba Tombol dan Menggerakan *Mouse* pada Android

Pada skenario ini dipaparkan tahap pengujian perangkat lunak untuk menggerakan dan menekan tombol *mouse* yaitu dilakukan pengujian dengan menggerakan dan menekan tombol

mouse seperti pada Tabel 5.11, dimana masukan yang digunakan adalah *touchpad* pada aplikasi *client* lalu dilihat hasilnya apakah sesuai dengan yang dihasilkan di komputer.

Tabel 5.11 Uji Coba Tombol dan Menggerakan *Mouse* pada Android

No	Gerakan pada Mouse	Hasil yang Keluar pada Komputer	Hasil
1	Menggerakan touchpad ke atas	Pointer bergerak ke atas	Sukses
2	Menggerakan touchpad ke bawah	Pointer bergerak ke bawah	Sukses
3	Menggerakan touchpad ke kanan	Pointer bergerak ke kanan	Sukses
4	Menggerakan touchpad ke kiri	Pointer bergerak ke kiri	Sukses
5	Menekan tombol L pada layar aplikasi	Klik kiri pada pointer	Sukses
6	Menekan tombol R pada layar aplikasi	Klik kanan pada pointer	Sukses
7	Menekan tombol V pada layar aplikasi	Scroll ke bawah	Sukses
8	Menekan tombol ^ pada layar aplikasi	Scroll ke atas	Sukses

Dapat dilihat dari tabel 5.11 bahwa tombol *mouse* yang ditekan dan digerakan pada Android sesuai dengan hasil yang keluar pada komputer.

5.2.3 Uji Coba Delay (*Latency*)

Delay atau *Latency* adalah jumlah waktu yang dibutuhkan paket data untuk berpindah di seluruh koneksi jaringan. Ketika sebuah paket sedang dikirim, ada "laten" waktu, ketika komputer yang mengirim paket menunggu konfirmasi bahwa paket telah diterima.

Pada uji coba kali ini akan beracuan pada log *delay* yang dihasilkan oleh aplikasi berupa file *Delay_log.txt*. Log *delay*

sendiri dihasilkan dari selisih waktu yang dibutuhkan oleh *command* mulai dikirimkan oleh *client* ke *server* sampai diproses oleh aplikasi *server*.

Berikut ini isi dari file Delay_log.txt.

```
=====
Start : 1438051881648 ms
End : 1438051881670 ms
Delay : 22 ms
Command : [G]-18,2
=====
```

```
=====
Start : 1438051881672 ms
End : 1438051881693 ms
Delay : 21 ms
Command : [G]-37,7
=====
```

```
=====
Start : 1438051883293 ms
End : 1438051883321 ms
Delay : 28 ms
Command : [G]24,15
=====
```

```
=====
Start : 1438051883322 ms
End : 1438051883357 ms
Delay : 35 ms
Command : [G]58,30
=====
```

```
=====
Start : 1438051883359 ms
End : 1438051883381 ms
Delay : 22 ms
Command : [G]76,37
=====
```

```
=====
Start : 1438051883383 ms
End : 1438051883404 ms
Delay : 21 ms
Command : [G]95,41
=====
```

```
=====
Start : 1438051885444 ms
End : 1438051885478 ms
Delay : 34 ms
Command : [M]L
=====
```

Start : 1438051894906 ms
 End : 1438051894944 ms
 Delay : 38 ms
 Command : [M]R

Start : 1438051918186 ms
 End : 1438051918222 ms
 Delay : 36 ms
 Command : [K]Q

Start : 1438051919115 ms
 End : 1438051919141 ms
 Delay : 26 ms
 Command : [K]W

Start : 1438051919550 ms
 End : 1438051919577 ms
 Delay : 27 ms
 Command : [K]E

Start : 1438051922018 ms
 End : 1438051922055 ms
 Delay : 37 ms
 Command : [K]I

Start : 1438051922367 ms
 End : 1438051922379 ms
 Delay : 12 ms
 Command : [K]2

Start : 1438051922751 ms
 End : 1438051922786 ms
 Delay : 35 ms
 Command : [K]3

Keterangan:

Command [K]	= menjalankan fungsi <i>keyboard</i>
Command [M]L	= menjalankan fungsi <i>mouse</i> klik kiri
Command [M]R	= menjalankan fungsi <i>mouse</i> klik kanan
Command [G]	= menjalankan fungsi <i>mouse gesture</i>

Tabel 5.12 Uji Coba Delay

No	Command	Delay (ms)
1	[G]-18,2	22
2	[G]-37,7	21
3	[G]24,15	28
4	[G]58,30	35
5	[G]76,37	22
6	[G]95,41	21
7	[M]L	34
8	[M]R	38
9	[K]Q	36
10	[K]W	26
11	[K]E	27
12	[K]1	37
13	[K]2	12
14	[K]3	35
Rata-rata delay		28,14

Dari log *delay* dibuat tabel *delay* seperti pada tabel 5.12 yang menghasilkan rata-rata *delay* 28,14 ms.

Sedangkan standar deviasi *delay* dihitung dengan rumus

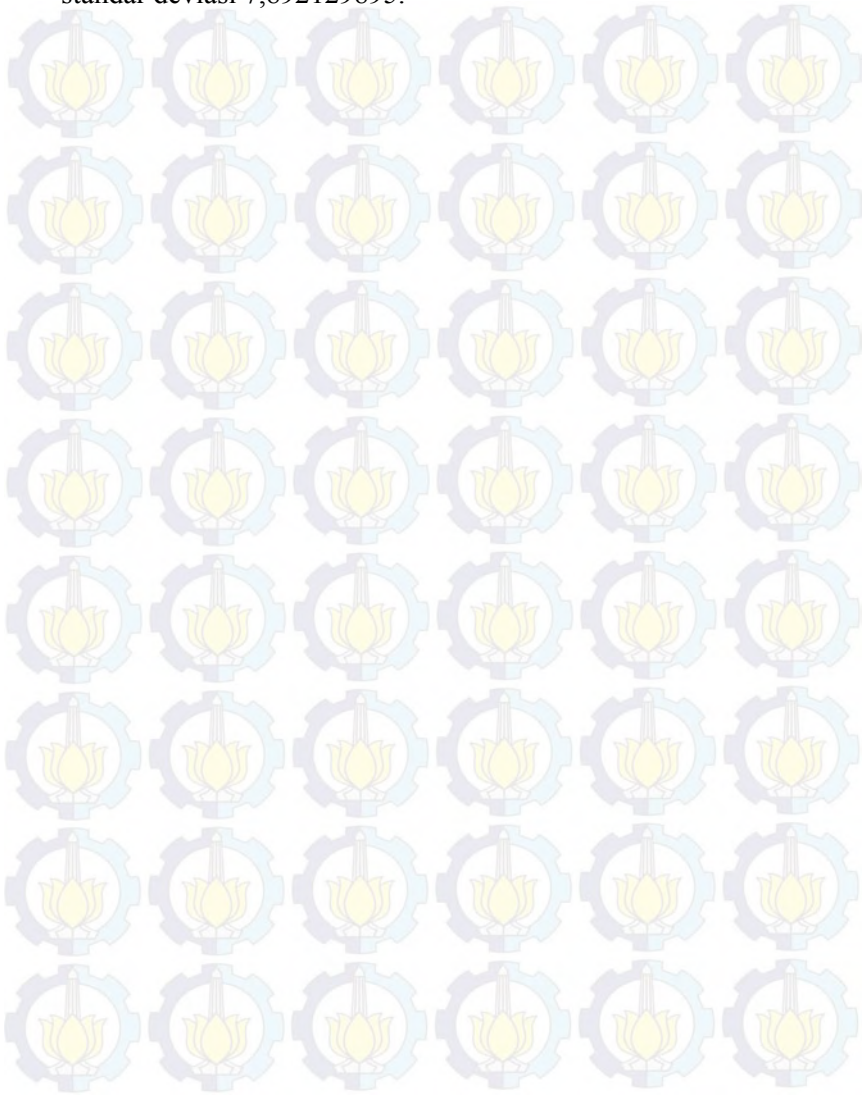
$$s = \sqrt{\frac{\sum (x_i - x)^2}{n - 1}}$$

x : merupakan rata-rata hasil *delay*

n : jumlah uji coba *delay* yang dilakukan

$$s = \sqrt{\frac{\sum (x_i - 28,14)^2}{14 - 1}} = 7,892129895$$

Pengujian *delay* pada pada aplikasi menghasilkan nilai standar deviasi 7,892129895.



BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Kesimpulan dari hasil uji coba dan evaluasi perangkat lunak yang dilakukan adalah sebagai berikut:

1. Otentikasi menggunakan *username*, *password*, dan *MAC address* pada aplikasi dapat berjalan dengan baik dan diimplementasikan dengan benar.
2. *Keyboard*, tombol angka, huruf, dan lain-lain sebagainya dapat berjalan dengan baik.
3. Fitur *multi-touch* untuk mengimplementasikan tombol khusus pada *keyboard* seperti *ctrl+a*, *ctrl+c*, *ctrl+v*, dan lain sebagainya dapat dijalankan, namun masih terbatas *multi-touch* dua jari (gabungan dua tombol *keyboard*). Karena untuk mengimplementasikan *multi-touch* lebih dari dua tombol membutuhkan *resource* yang sangat besar.
4. *Mouse touchpad* dapat dijalankan dengan baik, serta tombol mouse seperti klik kiri, klik kanan, dan *scroll* dapat berjalan dengan baik.

6.2 Saran

Saran untuk pengembangan aplikasi remote komputer berbasis android menggunakan Wifi dengan *Multi-touch* pada Tugas Akhir ini antara lain:

1. Untuk pengembangan aplikasi lebih lanjut disarankan untuk dapat dikembangkan di dalam *platform* Windows Phone, IOS, Linux, dan Mac OSX.

2. Untuk pengembangan aplikasi selanjutnya disarankan menggunakan *multi-touch* lebih dari dua jari sehingga fungsi-fungsi tombol khusus dapat diimplementasikan seperti ctrl+alt+delete dan lain-lain.
3. Penambahan tombol-tombol khusus seperti fn, windows, F1, F2, F3 dan lain-lain.



BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Kesimpulan dari hasil uji coba dan evaluasi perangkat lunak yang dilakukan adalah sebagai berikut:

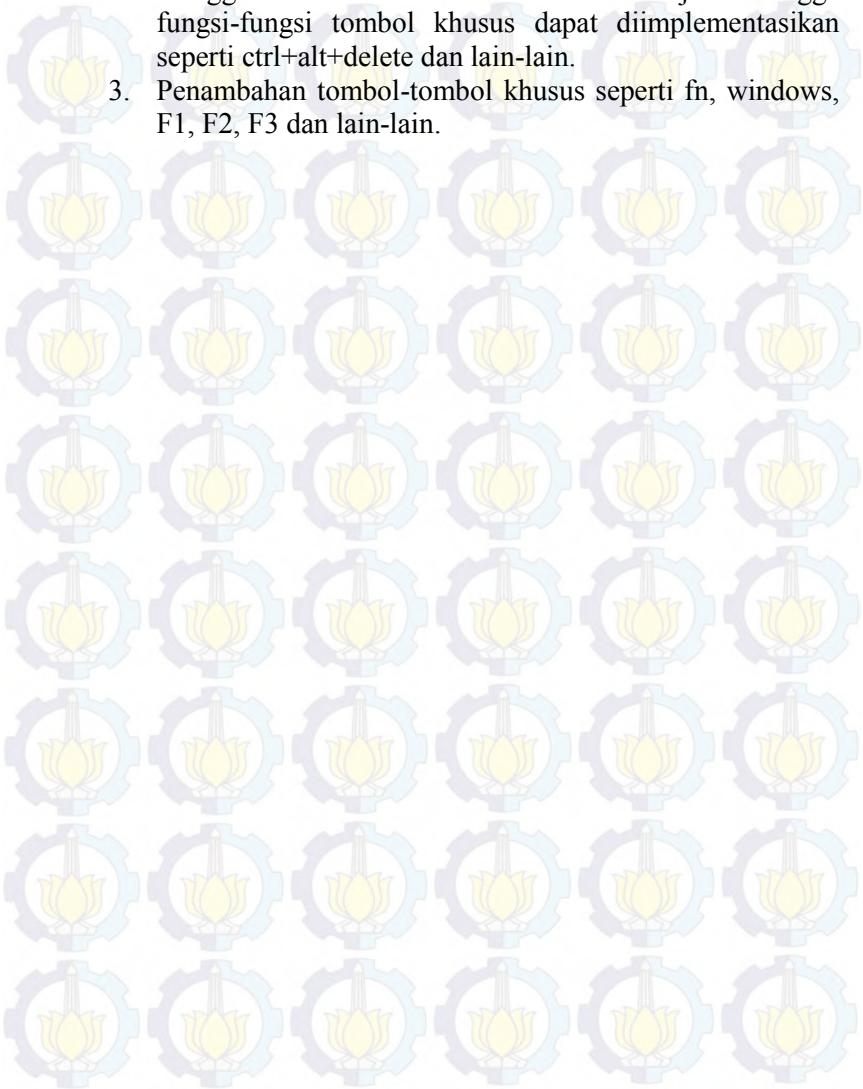
1. Otentikasi menggunakan *username*, *password*, dan *MAC address* pada aplikasi dapat berjalan dengan baik dan diimplementasikan dengan benar.
2. *Keyboard*, tombol angka, huruf, dan lain-lain sebagainya dapat berjalan dengan baik.
3. Fitur *multi-touch* untuk mengimplementasikan tombol khusus pada *keyboard* seperti *ctrl+a*, *ctrl+c*, *ctrl+v*, dan lain sebagainya dapat dijalankan, namun masih terbatas *multi-touch* dua jari (gabungan dua tombol *keyboard*). Karena untuk mengimplementasikan *multi-touch* lebih dari dua tombol membutuhkan *resource* yang sangat besar.
4. *Mouse touchpad* dapat dijalankan dengan baik, serta tombol mouse seperti klik kiri, klik kanan, dan *scroll* dapat berjalan dengan baik.

6.2 Saran

Saran untuk pengembangan aplikasi remote komputer berbasis android menggunakan Wifi dengan *Multi-touch* pada Tugas Akhir ini antara lain:

1. Untuk pengembangan aplikasi lebih lanjut disarankan untuk dapat dikembangkan di dalam *platform* Windows Phone, IOS, Linux, dan Mac OSX.

2. Untuk pengembangan aplikasi selanjutnya disarankan menggunakan *multi-touch* lebih dari dua jari sehingga fungsi-fungsi tombol khusus dapat diimplementasikan seperti ctrl+alt+delete dan lain-lain.
3. Penambahan tombol-tombol khusus seperti fn, windows, F1, F2, F3 dan lain-lain.



DAFTAR PUSTAKA

- [1] Shalahuddin M. 2009. *Belajar Pemrograman dengan Bahasa Pemrograman C++ dan Java*. Bandung: Informatika.
- [2] Nazruddin Safaat H. 2012. *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Informatika.
- [3] JARC Team, Modul Pengembangan Aplikasi Android, Seamolec Jakarta.
- [4] Wijayanto B. 2010. *Cara Jitu Pengelolaan Windows Dengan Remote Desktop Dan Administration*. Malang: Wahana Komputer.
- [5] Hantoro, Gunadi Dwi. 2010. *Wifi (Wireless LAN) Jaringan Komputer Tanpa Kabel*. Bandung: Informatika.
- [6] Zam, Efvy Zamidra. 2011. *Panduan Lengkap Membuat Jaringan Wireless*. Bandung: Elex Media.
- [7] Stevens M. 2007. *On Collisions for MD5*. <URL: <http://www.win.tue.nl/hashelash/OnCollisionsforMD-M.M.J.Stevens.pdf> diakses 25 Mei 2015>.
- [8] Pemrograman Android Dengan Eclipse. 2015. <URL: http://jack.lab.akprind.ac.id/doc/kuliah/13.2/13.2/pemrograman_mobile diakses 29 Mei 2015>.

BIODATA PENULIS



Mukhammad Syaifullah, biasa dipanggil Saiful, lahir di Sidoarjo pada tanggal 28 Agustus 1987, merupakan anak ke empat dari enam bersaudara. Penulis telah menempuh pendidikan mulai dari MI Tarbiyatul Akhlaq Taman Sidoarjo (1994-2000), SMP Nidhomuddin Taman Sidoarjo (2000-2002), MTs Sunan Kali Jaga Mojo Kediri (2002-2003), SMA Negeri 7 Kediri (2003-2006), dan pada tahun 2007 penulis meneruskan pendidikannya di Teknik Informatika ITS.

Di Teknik Informatika ITS, penulis mengambil bidang minat NCC (*Net Centric Computing*). Di sela-sela kesibukan dikampus, penulis yang hobi bermain bola serta gemar bermain *game Winning Eleven*, sangat suka menghabiskan waktu berjam-jam di depan komputer berkeliling dunia maya sekedar menambah wawasan di bidang IT dan menambah teman maya. Penulis juga hobi menonton film anime dan drama-drama asia terutama K-Drama dan J-Dorama. Penulis dapat dihubungi melalui email di ipunk.its@gmail.com.

Aplikasi Remote Komputer Berbasis Android Menggunakan Wifi dengan *Multi-touch*

M. Syaifullah, Radityo Anggoro, Henning Titi Ciptaningtyas
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
(ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail onngo@if.its.ac.id

Abstrak— *Smartphone* telah berkembang menjadi bagian yang penting bagi berbagai kalangan masyarakat. Dalam perkembangannya sekarang ini *smartphone* Android menjadi yang paling diminati. Android sendiri adalah *operating system* yang bersifat *open source*. Yang dimaksud *open source* bahwa OS Android adalah gratis, sehingga memungkinkan para programmer untuk mengembangkan atau membuat aplikasi berbasis Android.

Pada umumnya ponsel Android sudah dilengkapi dengan teknologi Wifi untuk kebutuhan wireless. Dengan memanfaatkan teknologi Wifi dan fitur layar sentuh yang dimiliki oleh semua ponsel Android dapat dibuat sebuah aplikasi pengganti *mouse* atau *touchpad* untuk mengendalikan komputer dari jarak jauh dengan Wifi sebagai penghubungnya.

Dalam artikel ini menjelaskan pembuatan aplikasi remote komputer dengan pemanfaatan layar sentuh *smartphone* berbasis Android sebagai pengganti *mouse* atau *touchpad* yang dilengkapi dengan fungsi *keyboard* pada komputer. Aplikasi yang dibuat dijalankan pada komputer untuk aplikasi *server* dan ponsel android untuk aplikasi *client*.

Aplikasi ini dapat mempermudah para penggunanya dalam melakukan presentasi dan mengontrol komputernya dari jarak jauh. Serta dengan pemanfaatan username, password, serta MAC address sehingga dapat mencegah pihak luar mengendalikan komputer pengguna.

Kata Kunci— *Smartphone*, *Android*, *Multi-touch*, *Remote Komputer*, *Wifi*.

I. PENDAHULUAN

Dalam dunia IT (*Information Technology*) saat sekarang ini, perkembangan teknologi dirasa sangat pesat. Dahulu orang yang mempunyai ponsel masih terbatas dan fitur-fiturnya hanya sekedar untuk mengirim pesan dan menelpon serta hanya kalangan menengah atas saja yang mempunyainya, akan tetapi saat ini hampir semua kalangan mempunyai ponsel dan fiturnya sendiri tidak hanya untuk menelpon tetapi juga bisa sekaligus mengambil gambar, mendengarkan musik, bermain game, dan masih banyak lagi.

Pada saat ini penggunaan sistem operasi pada ponsel bukan merupakan hal yang baru lagi. Dengan adanya sistem operasi pada ponsel, kita dapat memasukkan bermacam-macam aplikasi dibandingkan dengan ponsel yang tanpa sistem operasi. Beberapa tahun yang lalu sistem operasi ponsel yang terkenal adalah Symbian akan tetapi dominasi tersebut seakan runtuh dan digantikan oleh sistem operasi Android.

Android diperkenalkan oleh Google pada tahun 2005 dan didistribusikan pada tahun 2007. Pada umumnya perangkat *mobile* yang menggunakan sistem operasi Android menggunakan layar sentuh dan Android sendiri memang dirancang untuk perangkat *mobile* berbasis *touchscreen*. Perangkat *mobile* yang memakai sistem operasi Android rata-rata sudah terdapat teknologi Wifi didalamnya, dahulu ponsel yang terdapat fitur Wifi merupakan ponsel kelas atas, akan tetapi saat ini ponsel dengan harga dibawah 1 juta sudah bisa merasakan teknologi Wifi. Wifi sendiri merupakan teknologi untuk menghubungkan antara satu perangkat dengan perangkat lainnya tanpa menggunakan kabel sehingga menjadi lebih praktis.

Laptop zaman sekarang bisa dikatakan pasti memiliki fitur Wifi dan harga Wifi adaptor untuk komputer juga sudah murah, oleh karena itu, penulis mencoba membuat aplikasi yang memanfaatkan teknologi ini untuk menghubungkan antara komputer dengan perangkat *mobile* Android dan dalam hal ini penulis mencoba membuat aplikasi *remote* komputer via Wifi menggunakan Android.

Pada artikel ini, penulis mencoba mempermudah *user* dalam mengendalikan komputernya dari jarak jauh. Hal ini sangat berguna apabila *user* sedang presentasi di suatu ruangan tanpa didampingi seseorang sehingga *user* tidak perlu bolak-balik ke komputer untuk menggerakkan *mouse* ataupun menekan *keyboard*.

II. TINJAUAN PUSTAKA

A. Java

Java adalah bahasa pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Mereka lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional.

Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *bytecode* dan dapat dijalankan pada berbagai JVM (*Java Virtual Machine*). Java merupakan bahasa pemrograman yang bersifat umum, dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa sistem operasi yang berbeda, sehingga java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*".

B. Android

Android adalah sistem operasi untuk *mobile device* yang awalnya dikembangkan oleh Android Inc. Google Inc membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel. Pada saat perilis perdana Android, 5 November 2007, Android bersama *open handset alliance*, yaitu konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia, menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, google merilis kode-kode android di bawah lisensi apache. Android dibuat berdasarkan kernel Linux yang dimodifikasi. Aplikasi Android ditulis dengan bahasa Java, menggunakan Java Core Libraries. Aplikasi Android dijalankan di atas VM bernama Dalvik Virtual Machine.

Pada saat ini, sudah banyak vendor smartphone yang memproduksi berbasis android, hal ini terjadi karena android adalah sistem operasi yang open source sehingga bebas di distribusikan & dipakai oleh vendor manapun. Android itu sendiri sangat lengkap baik dari segi sistem operasi, aplikasi dan tool pengembangan, market aplikasi serta dukungan yang sangat tinggi dari komunitas open source didunia.

Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

C. MD5

MD5 merupakan fungsi hash satu arah yang didesain oleh Ronald Rivest dengan *hash value* 128 bit. Pada standard internet, MD5 telah dimanfaatkan pada bermacam-macam aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah file.

Algoritma MD5 secara garis besar adalah mengambil pesan yang mempunyai panjang variabel diubah menjadi „sidik jari“ atau „intisari pesan“ yang mempunyai panjang tetap yaitu 128 bit. „Sidik jari“ ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari „sidik jari“ MD5.

Message Digest 5 (MD5) adalah salah satu dari serangkaian algoritma *Message Digest* yang didesain oleh Professor Ronald Rivest dari MIT. Saat kerja analitik menunjukkan bahwa pendahulu MD5 -MD4- mulai tidak aman, MD5

kemudian di desain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

MD5 banyak digunakan pada bermacam-macam aplikasi termasuk SSL/TLS, IPsec dan protocol-protokol kriptografi lainnya. MD5 juga biasa digunakan pada implementasi *Timestamping Mechanism*, *Commitment Schemes*, dan aplikasi pengecekan integritas pada *online software*.

D. Wifi

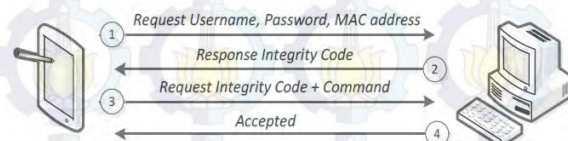
WiFi adalah singkatan dari *Wireless Fidelity* (sering ditulis dengan Wi-fi, WiFi, Wifi, wifi). Pengertian Wi-Fi adalah koneksi tanpa kabel yang menghubungkan jaringan komputer, seperti ponsel yang mempergunakan teknologi radio sehingga pengguna dapat melakukan transfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel baik di rumah maupun di kantor-kantor dan pusat-pusat bisnis. Oleh Karena itu banyak orang mengasosiasikan Wi-Fi dengan “Kebebasan” karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang meeting, kamar hotel, kampus, dan café-café yang bertanda “Wi-Fi Hot Spot”. Juga salah satu kelebihan dari Wi-Fi adalah kecepatannya yang beberapa kali lebih cepat dari modem kabel yang tercepat. Jadi pemakai Wi-Fi tidak lagi harus berada di dalam ruang kantor untuk bekerja.

Wi-Fi Alliance mendefinisikan Wi-Fi sebagai “produk jaringan wilayah lokal nirkabel (WLAN) apapun yang didasarkan pada standar *Institute of Electrical and Electronics Engineers* (IEEE) 802.11”. Meski begitu, karena kebanyakan WLAN zaman sekarang didasarkan pada standar tersebut, istilah “Wi-Fi” dipakai dalam bahasa Inggris umum sebagai sinonim “WLAN”.

Wifi mempunyai sejarah keamanan yang berubah. Sistem enkripsi pertamanya, WEP, terbukti mudah ditembus. Protokol berkualitas lebih tinggi lagi, WPA dan WPA2 kemudian ditambahkan, tetapi sebuah fitur opsional yang ditambahkan tahun 2007 bernama Wi-Fi Protected Setup (WPS), memiliki celah yang memungkinkan penyerang mendapatkan kata sandi WPA atau WPA2 router dari jarak jauh dalam beberapa jam saja. Sejumlah perusahaan menyarankan untuk mematikan fitur WPS. Wi-Fi Alliance sejak itu memperbarui rencana pengujian dan program sertifikasinya untuk menjamin semua peralatan yang baru disertifikasi kebal dari serangan AP PIN yang keras.

III. DESAIN DAN PERANCANGAN SISTEM

A. Arsitektur Umum Sistem



Gambar 1. Arsitektur Umum

Rancangan arsitektur dari sistem yang akan dibuat dapat digambarkan seperti pada Gambar 1. Proses pertama yaitu,

pengguna menghubungkan perangkat Android dengan komputer yang akan dikendalikan menggunakan Wifi, pada aplikasi komputer pengguna memasukkan *username*, *password*, *server port*, dan *MAC address* perangkat Android, pengguna juga bisa mengatur *mouse ratio*, lalu pengguna menjalankan *server* dan *robot*. Setelah itu, pengguna *login* di Android dengan memasukkan *username*, *password*, *server IP*, dan *server port*. Setelah *input* yang dimasukan sudah benar maka komputer akan mengirimkan kode integritas ke Android dan sesi terbentuk. Dengan kode integritas tersebut Android mengirim perintah ke komputer dan perintah dijalankan. Perintah yang dijalankan dapat dilihat pada daftar *log*.

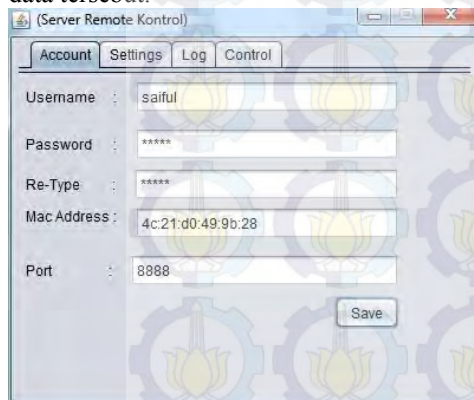
B. Desain Umum Sistem

Dalam artikel ini akan membangun suatu perangkat lunak pada komputer dan Android. Perangkat lunak yang merupakan sebuah aplikasi berbasis Android yang berfungsi sebagai alat untuk mengendalikan *keyboard* dan *mouse* yang ada di komputer. Aplikasi yang dibuat pada perangkat lunak ini menggunakan bahasa pemrograman Java.

IV. TAMPILAN APLIKASI

A. Tampilan Menu Utama Aplikasi Server

Gambar 2 menampilkan menu utama aplikasi *Server*. Pada menu ini pengguna mengisi *username*, *password*, *MAC address*, dan *port* yang nantinya akan disimpan dan disamakan dengan *username*, *password*, *MAC address*, *port* yang ada di Android. Setelah *textfield* diisi, *save* dipilih untuk menyimpan data tersebut.



Gambar 2. Menu utama aplikasi *Server* di komputer.

B. Tampilan Menu Utama Aplikasi Client

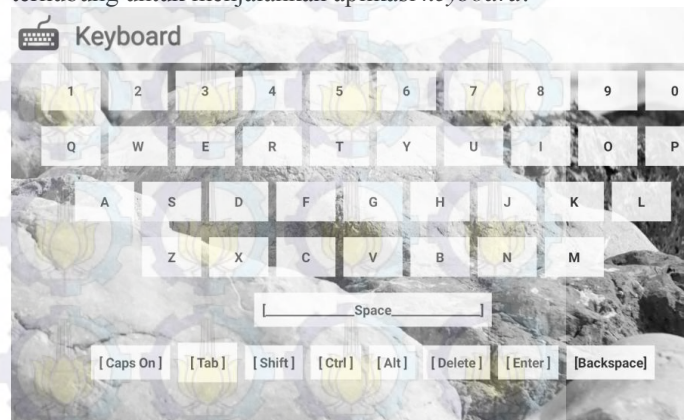
Gambar 3 menampilkan menu utama aplikasi *Client*. Pengguna memasukkan *username*, *password*, *server IP*, dan *server port* yang sesuai dengan yang ada pada aplikasi komputer.



Gambar 3. Tampilan menu utama aplikasi *Client* di Android.

C. Tampilan Keyboard Aplikasi Client

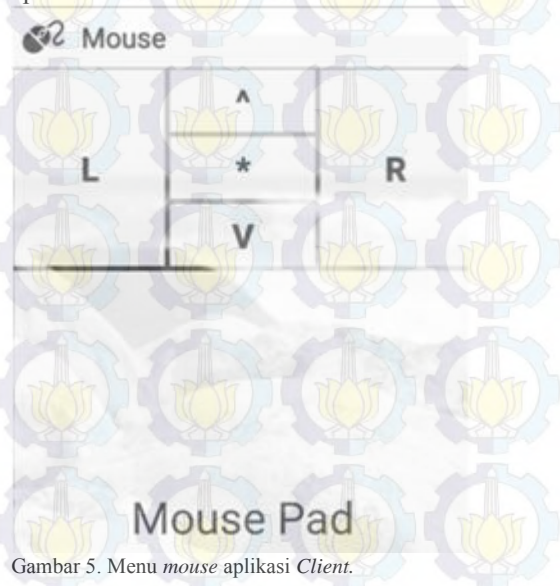
Gambar 4 menampilkan menu *keyboard* pada aplikasi. *keyboard* yang ditekan pada aplikasi Android akan sama hasilnya dengan yang keluar pada komputer. perangkat harus terhubung untuk menjalankan aplikasi *keyboard*.



Gambar 4. Menu *keyboard* aplikasi *Client*.

D. Tampilan Menu Mouse Aplikasi Client

Gambar 5 menampilkan menu *mouse* pada aplikasi *Client*. *Mouse* yang digerakan maupun yang ditekan pada Android akan sama hasilnya dengan tampilan *mouse* yang keluar pada komputer. Perangkat harus terhubung untuk menjalankan aplikasi.



Gambar 5. Menu *mouse* aplikasi *Client*.

V. UJI COBA

A. Uji Coba Login

Pada tahap pengujian perangkat lunak untuk melakukan pemeriksaan *login* yaitu:

- 1) Pengguna memasukkan *username*, *password*, *server IP*, *server port* yang sesuai dengan *server*.
- 2) Pengguna memilih *connect*.
- 3) Perangkat terhubung, apabila ada bagian yang salah, aplikasi akan meminta pemeriksaan kembali data yang dimasukkan.

Dari Tabel 1 terlihat bahwa untuk *login* ke *server* dari Android diperlukan masukan data yang sama baik dari sisi *client* maupun dari sisi *server*, apabila ada salah satu saja yang berbeda maka koneksi tidak dapat terhubung. Pada tahap pemeriksaan *login*, setelah *server* dan perangkat Android terhubung dengan Wifi, *server* akan mengecek data yang ada pada aplikasi Android apakah sama dengan data yang ada di *server*, jika sama koneksi tersambung, jika tidak maka aplikasi akan meminta untuk melakukan pemeriksaan ulang.

Tabel 1. Uji coba *login*

No	Input	Server	Client	Hasil
1	Username	Saiful	Saiful	Sukses
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
2	Username pada client tidak cocok dengan server			Gagal
	Username	Saiful	ipunk	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
3	Password pada client tidak cocok dengan server			Gagal
	Username	Saiful	saiful	
	Password	11111	22222	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
4	MAC Address pada server tidak cocok dengan client			Gagal
	Username	Saiful	saiful	
	Password	11111	11111	
	MAC Address	00:11:22:C4:C4:C3	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
5	IP Server pada client tidak cocok dengan server			Gagal
	Username	Saiful	saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.123	
6	Port pada client tidak cocok dengan server			Gagal
	Username	Saiful	saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	9999	

B. Uji Coba Keyboard

Pada tahap ini dilakukan pengujian kebenaran tombol *keyboard* yang dijalankan seperti pada Tabel 5.8, dimana masukan yang digunakan adalah tombol aplikasi *keyboard* pada Android dan hasilnya berupa tampilan pada komputer.

Tabel 2. Uji coba *keyboard*

No	Tombol Keyboard yang Ditekan	Hasil yang Keluar pada Komputer	Hasil
1	abcdefghijklmnopqrstuvwxyz	Mengetik huruf abcdefghijklmnopqrstuvwxyz	Sukses
2	1234567890	Mengetik angka 1234567890	Sukses
3	[Enter]	Pada layar teks kursor turun 1 baris	Sukses
4	[Tab]	Memindahkan kursor dalam 1 tabulasi ke arah kanan	Sukses
5	[Space]	Membuat jarak 1 karakter antar kata	Sukses
6	[Backspace]	Menghapus 1 karakter terakhir yang telah diketik	Sukses
7	[Delete]	File yang dipilih dihapus	Sukses
8	[Caps lock]	Sesuai dengan kondisi Caps lock	Sukses
9	[Ctrl] + a	Menjalankan perintah "Select all"	Sukses
10	[Ctrl] + c	Menjalankan perintah "Copy"	Sukses
11	[Ctrl] + v	Menjalankan perintah "Paste"	Sukses
12	[Ctrl] + x	Menjalankan perintah "Cut"	Sukses
13	[Ctrl] + z	Menjalankan perintah "Undo"	Sukses

C. Uji Coba Mouse

Pada skenario ini dipaparkan tahap pengujian perangkat lunak untuk menggerakkan dan menekan tombol *mouse* yaitu dilakukan pengujian dengan menggerakkan dan menekan tombol *mouse* seperti pada Tabel 3, dimana masukan yang digunakan adalah *touchpad* pada aplikasi *client* lalu dilihat hasilnya apakah sesuai dengan yang dihasilkan di komputer.

Tabel 3. Uji coba *mouse*

No	Gerakan pada Mouse	Hasil yang Keluar pada Komputer	Hasil
1	Menggerakkan touchpad ke atas	Pointer bergerak ke atas	Sukses
2	Menggerakkan touchpad ke bawah	Pointer bergerak ke bawah	Sukses
3	Menggerakkan touchpad ke kanan	Pointer bergerak ke kanan	Sukses
4	Menggerakkan touchpad ke kiri	Pointer bergerak ke kiri	Sukses
5	Menekan tombol L pada layar aplikasi	Klik kiri pada pointer	Sukses
6	Menekan tombol R pada layar aplikasi	Klik kanan pada pointer	Sukses
7	Menekan tombol V pada layar aplikasi	Scroll ke bawah	Sukses
8	Menekan tombol ^ pada layar aplikasi	Scroll ke atas	Sukses

D. Uji Coba Delay (Latency)

Delay atau *Latency* adalah jumlah waktu yang dibutuhkan paket data untuk berpindah di seluruh koneksi jaringan. Ketika sebuah paket sedang dikirim,

ada "laten" waktu, ketika komputer yang mengirim paket menunggu konfirmasi bahwa paket telah diterima.

Pada uji coba ini akan beracuan pada log *delay* yang dihasilkan oleh aplikasi berupa file *Delay_log.txt*. Log *delay* sendiri dihasilkan dari selisih waktu yang dibutuhkan oleh *command* mulai dikirimkan oleh *client* ke *server* sampai diproses oleh aplikasi *server*.

Tabel 4 Uji Coba Delay

No	Command	Delay (ms)
1	[G]-18,2	22
2	[G]-37,7	21
3	[G]24,15	28
4	[G]58,30	35
5	[G]76,37	22
6	[G]95,41	21
7	[M]L	34
8	[M]R	38
9	[K]Q	36
10	[K]W	26
11	[K]E	27
12	[K]I	37
13	[K]2	12
14	[K]3	35
Rata-rata delay		28,14

Dari log *delay* dibuat tabel *delay* seperti pada tabel 4 yang menghasilkan rata-rata *delay* 28,14 ms.

Sedangkan standar deviasi *delay* dihitung dengan rumus:

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

\bar{x} : merupakan rata-rata hasil *delay*

n : jumlah uji coba *delay* yang dilakukan

$$s = \sqrt{\frac{\sum (x_i - 28,14)^2}{14 - 1}} = 7,892129895$$

Pengujian *delay* pada aplikasi menghasilkan nilai standar deviasi 7,892129895.

VI. KESIMPULAN DAN SARAN

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

- 1) Otentikasi menggunakan *username*, *password*, dan *MAC address* pada aplikasi dapat berjalan dengan baik dan diimplementasikan dengan benar.
- 2) *Keyboard*, tombol angka, huruf, dan lain-lain sebagainya dapat berjalan dengan baik.
- 3) Fitur *multi-touch* untuk mengimplementasikan tombol khusus pada *keyboard* seperti *ctrl+a*, *ctrl+c*, *ctrl+v*, dan lain sebagainya dapat dijalankan, namun masih terbatas

multi-touch dua jari (gabungan dua tombol *keyboard*). Karena untuk mengimplementasikan *multi-touch* lebih dari dua tombol membutuhkan *resource* yang sangat besar.

- 4) *Mouse touchpad* dapat dijalankan dengan baik, serta tombol mouse seperti klik kiri, klik kanan, dan *scroll* dapat berjalan dengan baik.

Saran untuk pengembangan aplikasi remote komputer berbasis android menggunakan Wifi dengan *Multi-touch* pada artikel ini antara lain:

- 1) Untuk pengembangan aplikasi lebih lanjut disarankan untuk dapat dikembangkan di dalam *platform* Windows Phone, IOS, Linux, dan Mac OSX.
- 2) Untuk pengembangan aplikasi selanjutnya disarankan menggunakan *multi-touch* lebih dari dua jari sehingga fungsi-fungsi tombol khusus dapat diimplementasikan seperti *ctrl+alt+delete* dan lain-lain.
- 3) Penambahan tombol-tombol khusus seperti *fn*, *windows*, *F1*, *F2*, *F3* dan lain-lain.

UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Allah SWT atas limpahan rahmat dan hidayah-Nya dan terima kasih kepada Orang tua dan teman-teman tercinta yang telah memberikan dukungan dan semangat kepada penulis.

DAFTAR PUSTAKA

- [1] Shalahuddin M. 2009. *Belajar Pemrograman dengan Bahasa Pemrograman C++ dan Java*. Bandung: Informatika.
- [2] Nazruddin Safaat H. 2012. *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Informatika.
- [3] JARC Team, *Modul Pengembangan Aplikasi Android*, Seamolec Jakarta.
- [4] Wijayanto B. 2010. *Cara Jitu Pengelolaan Windows Dengan Remote Desktop Dan Administration*. Malang: Wahana Komputer.
- [5] Hantoro, Gunadi Dwi. 2010. *Wifi (Wireless LAN) Jaringan Komputer Tanpa Kabel*. Bandung: Informatika.
- [6] Zam, Efvy Zamidra. 2011. *Panduan Lengkap Membuat Jaringan Wireless*. Bandung: Elex Media.
- [7] Stevens M. 2007. *On Collisions for MD5*. <URL: <http://www.win.tue.nl/hashclash/OnCollisionsforMD-M.M.J.Stevens.pdf> diakses 25 Mei 2015>.
- [8] Pemrograman Android Dengan Eclipse. 2015. <URL: http://jack.lab.akprind.ac.id/doc/kuliah/13.2/13.2/pemrograman_mobile diakses 29 Mei 2015>.



ITS
Institut
Teknologi
Sepuluh Nopember

PRESENTASI TUGAS AKHIR

APLIKASI REMOTE KOMPUTER BERBASIS ANDROID MENGGUNAKAN WIFI DENGAN MULTI-TOUCH

Oleh:

M. Syaifullah
5107100159

Dosen Pembimbing 1:

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

Dosen Pembimbing 2:

Henning Titi Ciptaningtyas, S.Kom., M.Kom.

::LATAR BELAKANG(1)::

- Pengguna *smartphone* yang semakin meningkat dari tahun ke tahun.

Sebuah laporan dari Emarketer menyatakan bahwa akan terdapat 2 miliar pengguna *smartphone* aktif di seluruh dunia pada tahun 2016.

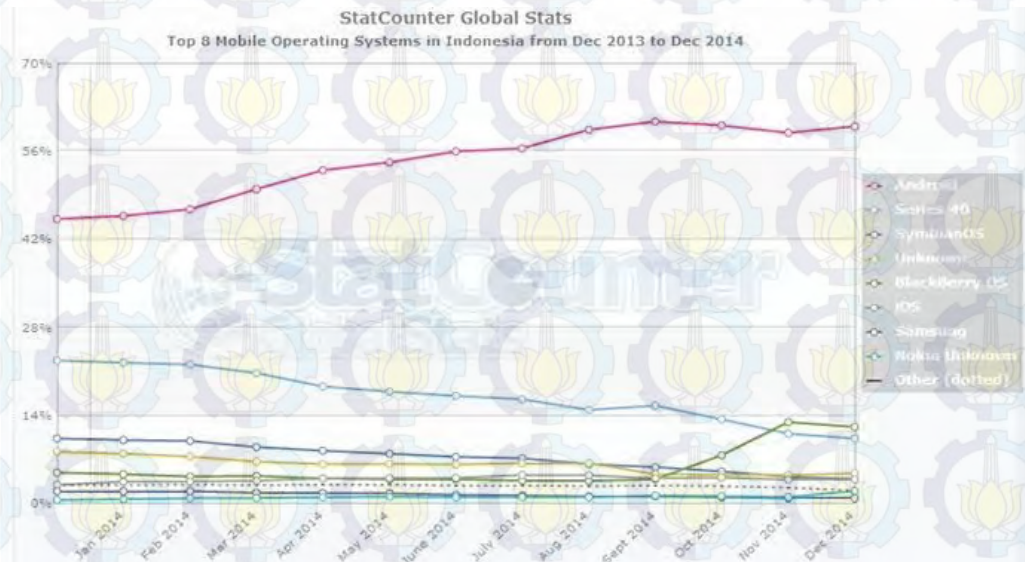
Di Indonesia sendiri akan melampaui 100 juta pengguna *smartphone* aktif pada tahun 2018.



:::LATAR BELAKANG(2):::

- *Smartphone* Android paling diminati dikalangan masyarakat.

Berdasarkan laporan StatCounter, selama tahun 2014 ternyata Android merupakan sistem operasi yang mendominasi peredaran smartphone di Indonesia dengan pembagian pasar sebesar 60 persen.



::LATAR BELAKANG(3)::

- *Smartphone* Android sudah menggunakan layar sentuh.
- Umumnya ponsel Android sudah dilengkapi dengan teknologi Wifi.
- Laptop sekarang pada umumnya pasti memiliki fitur Wifi.
- Dengan memanfaatkan teknologi Wifi dan fitur layar sentuh yang dimiliki ponsel Android, dalam Tugas Akhir ini dibuat aplikasi pengganti *mouse* atau *touchpad* dan dilengkapi dengan fungsi *keyboard* untuk mengendalikan komputer dari jarak jauh dengan Wifi sebagai penghubungnya.



.:TUJUAN:.

- Tujuan dari pengerjaan tugas akhir ini adalah untuk mengembangkan kemampuan perangkat *mobile* sehingga dapat digunakan untuk mengontrol komputer dari jarak jauh.
- Menggantikan fungsi *keyboard* dan *mouse* yang ada pada komputer.



::RUMUSAN MASALAH::

- WIFI -> Menghubungkan perangkat Android dengan komputer menggunakan Wifi.
- Sinkronisasi -> Komputer mengenal perangkat Android sebagai *keyboard* dan *mouse* yang akan digunakan.
- Multi-touch -> Menjadikan interface aplikasi dengan fitur *multi-touch* sehingga fungsi-fungsi tombol khusus pada *keyboard* seperti ctrl+a, ctrl+c, ctrl+v, dan lain sebagainya dapat diimplementasikan.



::BATASAN MASALAH::

- OS -> Windows dan Android
- Menggantikan fungsi *mouse* dan *keyboard* pada komputer.
- Wifi sebagai penghubung.
- Otentifikasi *username*, *password* serta *MAC address* untuk mengamankan jaringan.
- Fitur *multi-touch* untuk mengimplementasikan tombol khusus pada *keyboard* seperti ctrl+a, ctrl+c, ctrl+v, dan lain sebagainya.



Arsitektur Umum

..:RANCANGAN(1):.

Pengguna menghubungkan perangkat Android dengan komputer yang akan dikendalikan menggunakan Wifi. Pada aplikasi komputer pengguna memasukkan *username*, *password*, *server port*, dan *MAC address* perangkat Android, pengguna juga bisa mengatur *mouse ratio*, lalu menjalankan *server* dan *robot*.

Setelah itu, pengguna *login* di Android dengan memasukkan *username*, *password*, *server IP*, dan *server port*. Setelah *input* yang dimasukan sudah benar maka komputer akan mengirimkan kode integritas ke Android. Dengan kode integritas tersebut Android mengirim perintah ke komputer dan perintah dijalankan.

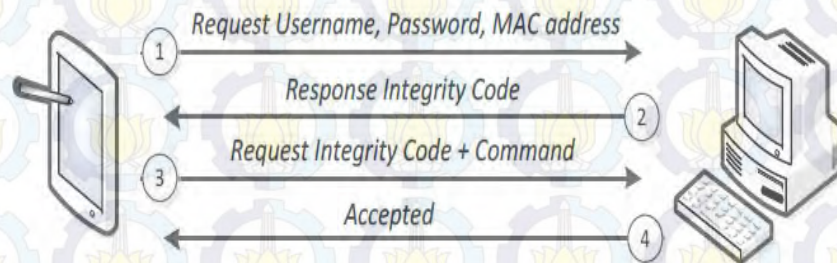


Diagram Kasus

..:RANCANGAN(2):.

Pada perancangan diagram kasus menggambarkan fungsionalitas sistem beserta aktor yang terlibat.

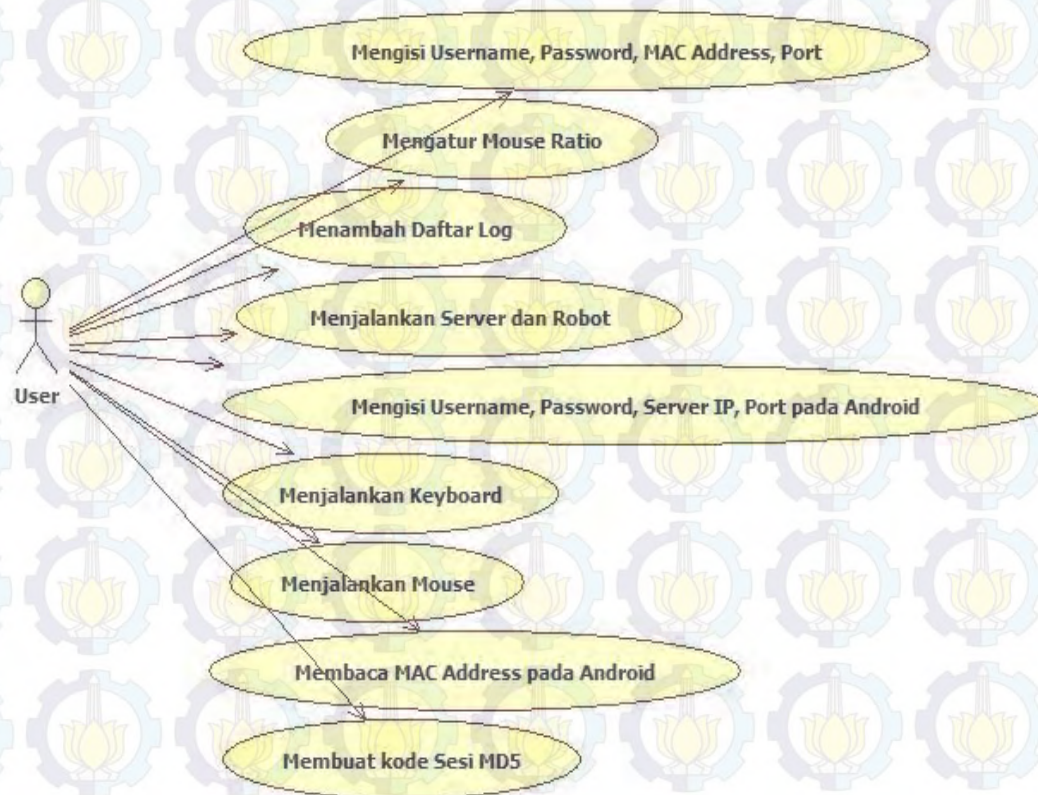


Diagram Alir Proses Pengisian *Username*, *Password*, *MAC Address*, *Port* pada Server

Proses diawali dengan memilih menu *account* pada halaman utama. Kemudian akan tampil *form account*. *User* memasukkan *username*, *password*, *MAC address*, dan *port*. Data tersebut akan disimpan di dalam data *account*.

.:Diagram Alir(1):.

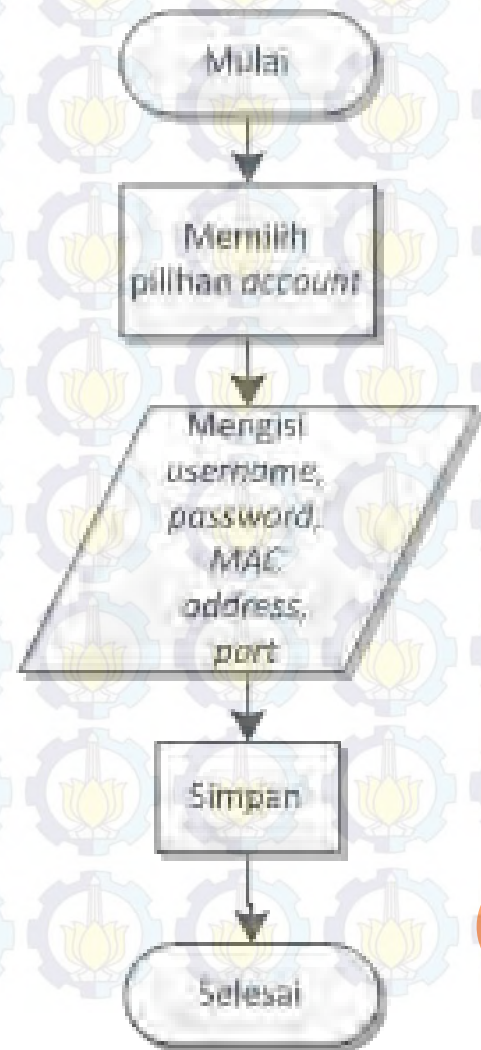


Diagram Alir Proses Pengisian *Username*, *Password*, *Server IP*, *Server Port* pada Android

Proses pengisian *username*, *password*, *server IP*, *server port* pada Android. Proses diawali dengan memasukkan *username*, *password*, *server IP*, *server port* yang sesuai pada aplikasi di Android. Kemudian *user* memilih *connect* untuk menghubungkan antara perangkat Android dengan *server*.

.:Diagram Alir(2):.

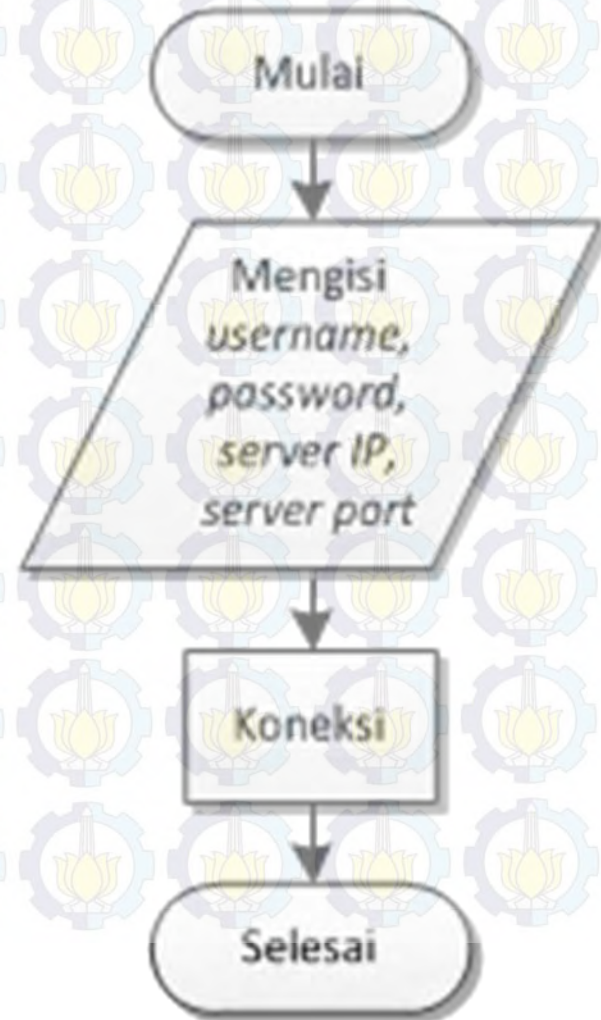
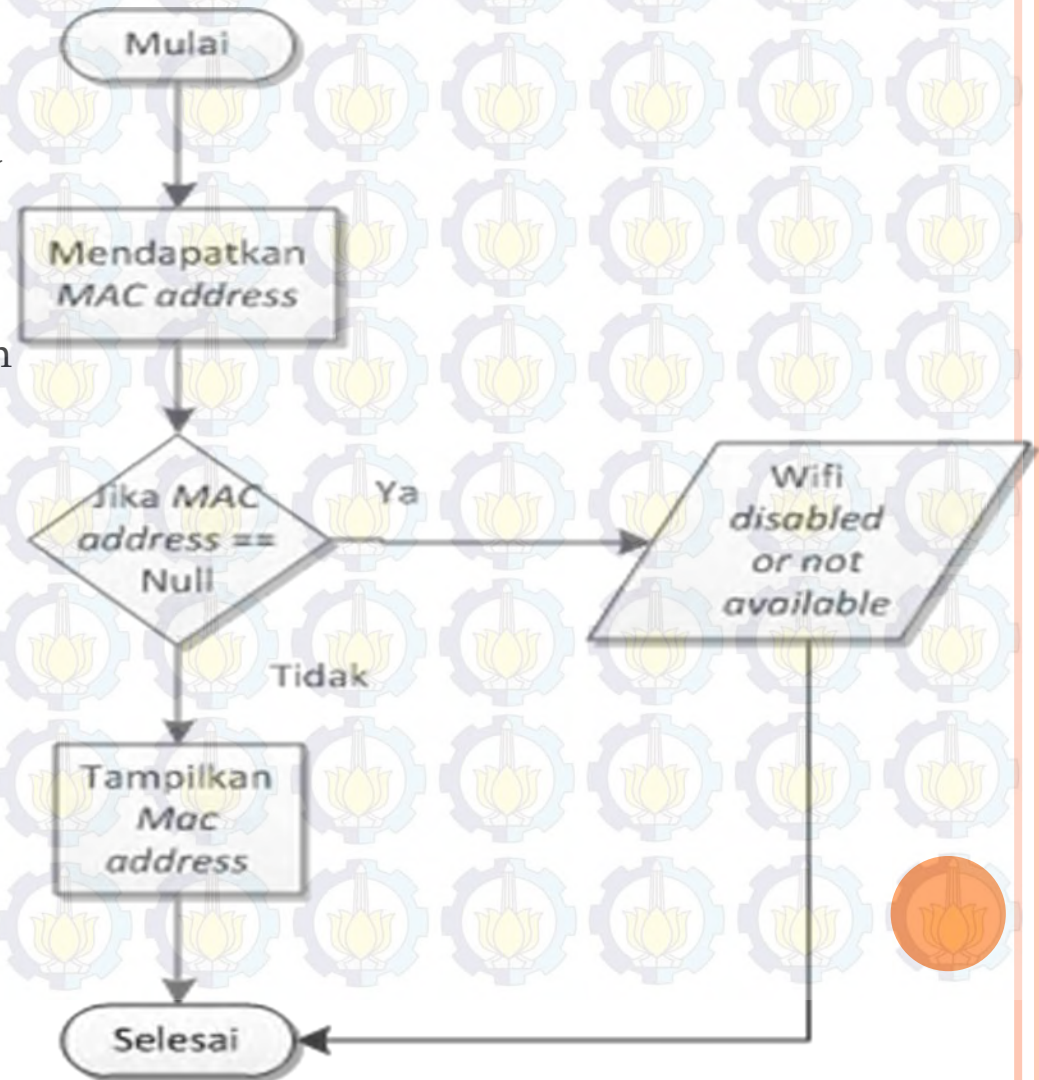


Diagram Alir Proses Pembacaan *MAC Address* pada Android

Proses pembacaan *MAC address* dilakukan saat membuka aplikasi *Client* pada Android. Aplikasi langsung mendeteksi *MAC address* dari perangkat tersebut, jika Wifi tidak aktif akan diberitahukan bahwa tidak ada perangkat Wifi atau Wifi tidak aktif.

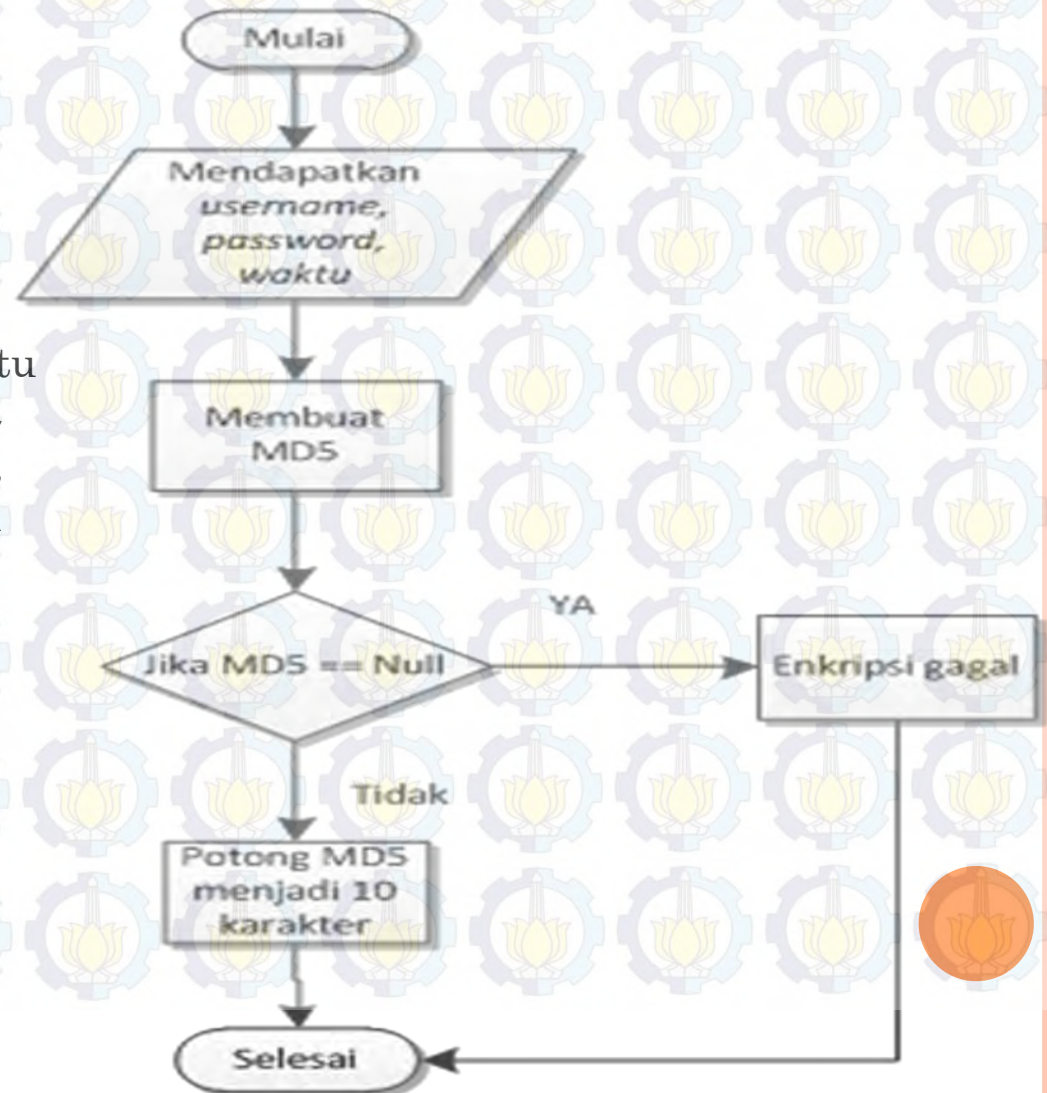


.:Diagram Alir(3):.

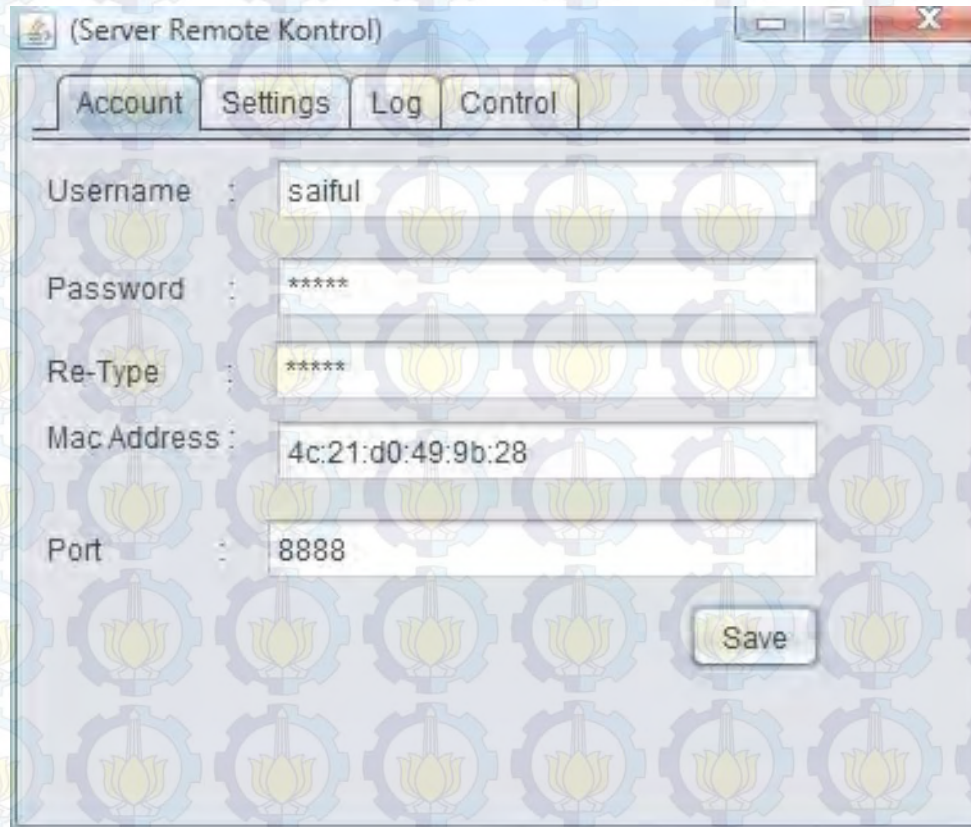
::Diagram Alir(4)::

Diagram Alir Proses Pembuatan Sesi MD5

Proses berlangsung ketika pengguna memasukkan *username*, *password*, *MAC address*, dan *port* kemudian menyimpan pengaturan *account* tersebut, saat *server* dijalankan (*run*) maka saat itu *server* akan membuat *session code* dengan dasar *username*, *password*, *MAC address*, dan waktu.



Pemasukan Username, Password, MAC Address, dan Server Port pada Server .: IMPLEMENTASI(1):.

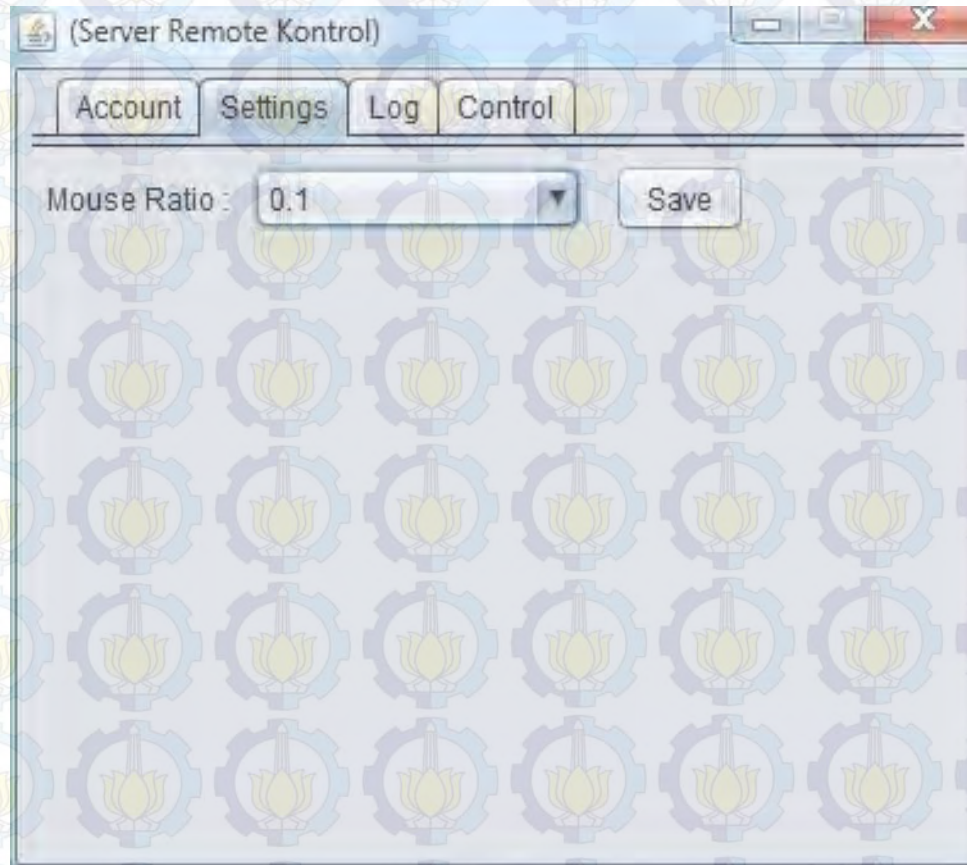


The screenshot shows a window titled "(Server Remote Kontrol)" with a standard Windows-style title bar. Inside the window, there are four tabs: "Account", "Settings", "Log", and "Control". The "Account" tab is currently selected. Below the tabs, there are five input fields, each preceded by a label and a colon. The fields are: "Username" with the value "saiful", "Password" with "*****", "Re-Type" with "*****", "Mac Address" with "4c:21:d0:49:9b:28", and "Port" with "8888". A "Save" button is located at the bottom right of the form area.

Field	Value
Username	saiful
Password	*****
Re-Type	*****
Mac Address	4c:21:d0:49:9b:28
Port	8888

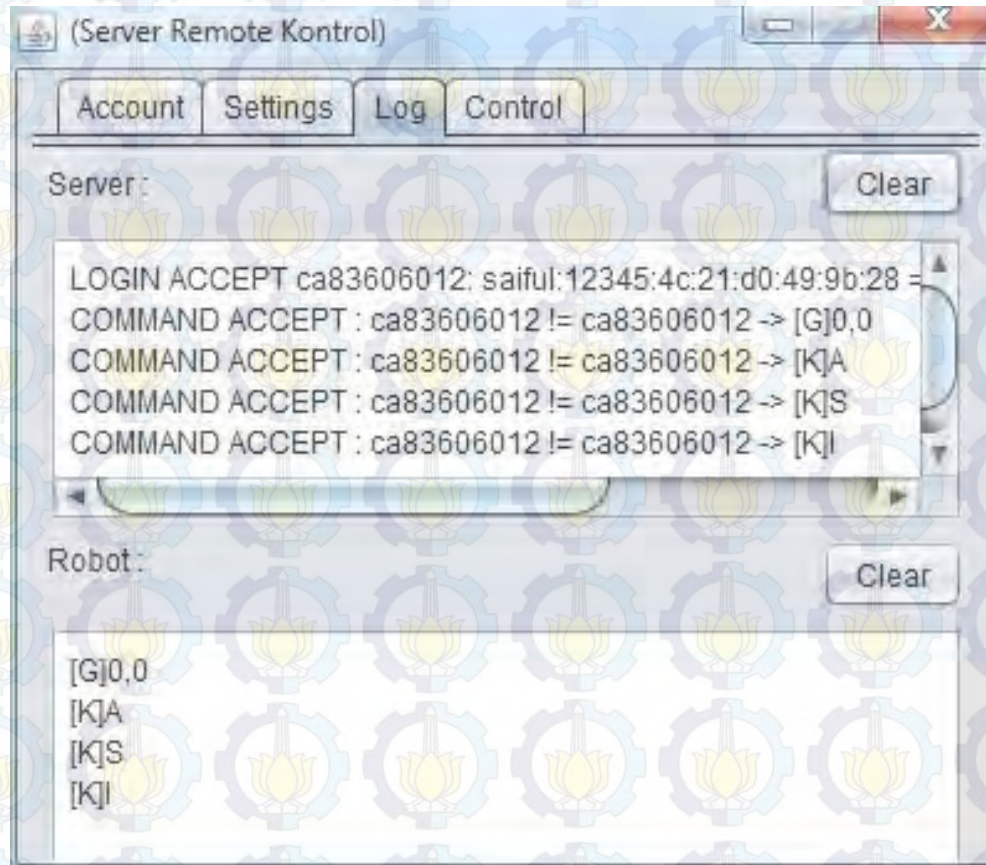


Menu Setting pada Server .: IMPLEMENTASI(2):.



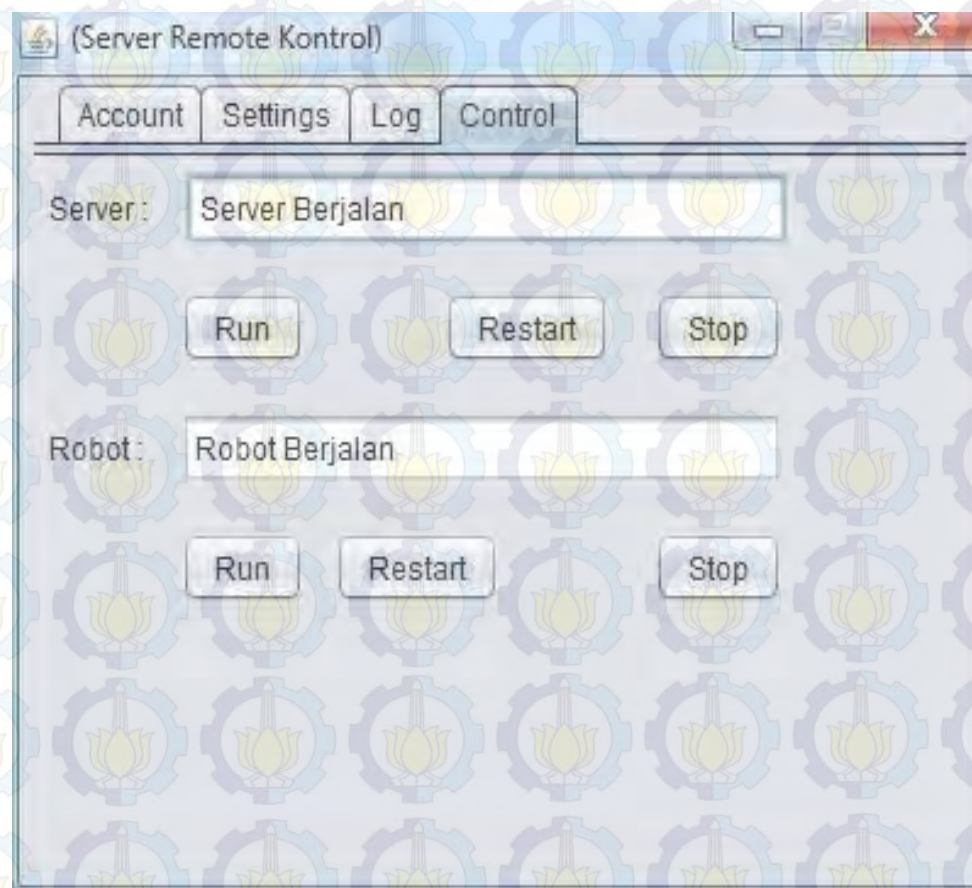
Menu Log pada Server

.: IMPLEMENTASI(3):.




.: IMPLEMENTASI(4):.

Menu Control pada Server



Pemasukan Username, Password,
Server IP, Server Port pada Android

.: IMPLEMENTASI(5):.

 **Hubungkan**

Username	saiful
Password
Mac Address	4c:21:d0:49:9b:28
IP Computer	192.168.43.141
Port	8888

Hubungkan

Keterangan

isi username, password, IP Address computer yang ingin di remote, beserta portnya kemudian klik hubungan

Login Berhasil, Sekarang anda bisa menggunakan keyboard dan mouse



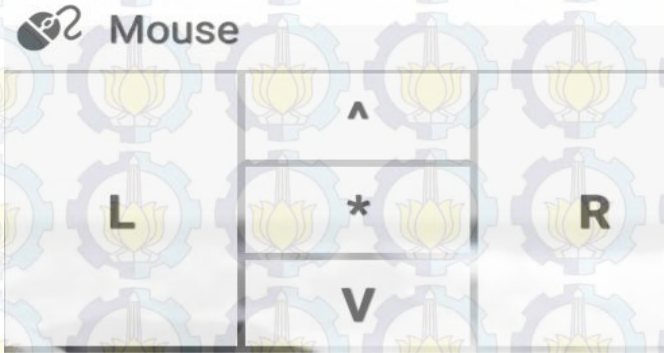
Keyboard pada Android

: IMPLEMENTASI(6):.



Mouse pada Android

.: IMPLEMENTASI(7):.



Mouse Pad



.:UJI COBA APLIKASI:.

Skenario uji coba aplikasi Tugas Akhir ini terdiri dari 2 uji coba:

- **Uji Coba Fungsionalitas**

Uji coba fungsionalitas program dilakukan untuk menguji fungsi dari tiap kontrol untuk setiap fitur yang ada pada aplikasi.

- **Uji Coba Kebenaran *Login, Mouse, Keyboard***

Uji coba ini dilakukan untuk menguji kebenaran *login* pada aplikasi untuk otentikasi, serta *keyboard* dan *mouse* sebagai aplikasi penggerak komputer.



.:Uji Coba Fungsionalitas(1):.

Uji Coba Pengisian *Username, Password, MAC Address*, dan *Port*

Kode Diagram Kasus	UJI-01
Nama uji coba	Uji coba pengisian username, password, MAC address, dan port.
Tujuan uji coba	Menguji fitur pengisian username, password, MAC address, dan port.
Skenario	Pengguna mengisi username, password, MAC address, dan port.
Kondisi awal	Perangkat berada pada menu account aplikasi Server.
Kondisi akhir	Perangkat berada pada menu account aplikasi Server.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(2):.

Uji Coba Pengaturan *Mouse Ratio*

Kode Diagram Kasus	UJI-02
Nama uji coba	Uji coba pengaturan mouse ratio.
Tujuan uji coba	Menguji perubahan mouse ratio pada aplikasi Server.
Skenario	Pengguna mengganti mouse ratio pada aplikasi Server.
Kondisi awal	Perangkat berada pada menu setting aplikasi Server.
Kondisi akhir	Perangkat berada pada menu setting aplikasi Server.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(3):.

Uji Coba Penambahan Daftar Log

Kode Diagram Kasus	UJI-03
Nama uji coba	Uji coba penambahan dan penghapusan log.
Tujuan uji coba	Menguji fitur penambahan dan penghapusan log pada aplikasi Server.
Skenario	Pengguna menjalankan aplikasi dan proses tersebut tercatat dalam daftar log dan log tersebut dihapus.
Kondisi awal	Perangkat berada pada pilihan menu log aplikasi Server.
Kondisi akhir	Perangkat berada pada menu log aplikasi Server dengan log server dan log robot mencatat proses yang berlangsung.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(4):.

Uji Coba Menjalankan *Server* dan *Robot*

Kode Diagram Kasus	UJI-04
Nama uji coba	Uji coba menjalankan server dan robot.
Tujuan uji coba	Menguji server dan robot
Skenario	Pengguna menjalankan server dan robot dan menghentikannya setelah mengisi textfield yang tersedia pada menu account.
Kondisi awal	Perangkat berada pada pilihan menu control pada aplikasi Server.
Kondisi akhir	Perangkat berada pada pilihan menu control pada aplikasi Server.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(5):.

Uji Coba Pengisian *Username, Password, Server IP, Server Port* pada Android

Kode Diagram Kasus	UJI-05
Nama uji coba	Uji coba pengisian username, password, server IP, server port pada Android.
Tujuan uji coba	Menguji otentikasi aplikasi Android dengan aplikasi yang ada pada komputer.
Skenario	Pengguna mengisi username, password, server IP, server port yang sesuai dan yang berbeda dengan aplikasi komputer.
Kondisi awal	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu Connect pada perangkat Android dengan status berhasil atau gagal terhubung.
Hasil uji coba	Berhasil dan gagal.



.:Uji Coba Fungsionalitas(6):.

Uji Coba Menjalankan *Keyboard*

Kode Diagram Kasus	UJI-06
Nama uji coba	Uji coba menjalankan keyboard.
Tujuan uji coba	Menguji kesamaan hasil antara keyboard yang ditekan pada aplikasi Android dengan hasil yang keluar pada komputer.
Skenario	Pengguna menjalankan keyboard pada aplikasi Android setelah login berhasil.
Kondisi awal	Perangkat berada pada pilihan menu keyboard pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu keyboard pada perangkat Android.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(7):.

Uji Coba Menjalankan *Mouse*

Kode Diagram Kasus	UJI-07
Nama uji coba	Uji coba menjalankan mouse.
Tujuan uji coba	Menguji kesamaan hasil antara mouse yang dijalankan pada aplikasi Android dengan hasil yang keluar pada komputer.
Skenario	Pengguna menjalankan mouse pada aplikasi Android setelah login berhasil.
Kondisi awal	Perangkat berada pada pilihan menu mouse pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu mouse pada perangkat Android.
Hasil uji coba	Berhasil.



.:Uji Coba Fungsionalitas(8):.

Uji Coba Membaca *MAC Address* pada Android

Kode Diagram Kasus	UJI-08
Nama uji coba	Uji coba membaca MAC address Pada Android.
Tujuan uji coba	Menguji kesamaan hasil antara MAC address pada perangkat yang sebenarnya dengan MAC address yang tampil pada aplikasi.
Skenario	Pengguna menjalankan aplikasi setelah menghidupkan Wifi.
Kondisi awal	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Kondisi akhir	Perangkat berada pada pilihan menu Connect pada perangkat Android.
Hasil uji coba	Berhasil.



::Uji Coba Kebenaran(1)::

Uji Coba pada Login

No	Input	Server	Client	Hasil
1	Username	saiful	saiful	Sukses
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
2	Username pada client tidak cocok dengan server			Gagal
	Username	saiful	ipunk	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
3	Password pada client tidak cocok dengan server			Gagal
	Username	saiful	saiful	
	Password	11111	22222	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	



::Uji Coba Kebenaran(1)::

Uji Coba pada Login

4	MAC Address pada server tidak cocok dengan client			Gagal
	Username	saiful	saiful	
	Password	11111	11111	
	MAC Address	00:11:22:C4:C4:C3	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	8888	
5	IP Server pada client tidak cocok dengan server			Gagal
	Username	saiful	saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.123	
	Port	8888	8888	
6	Port pada client tidak cocok dengan server			Gagal
	Username	saiful	saiful	
	Password	11111	11111	
	MAC Address	4c:21:d0:49:9b:28	4c:21:d0:49:9b:28	
	IP Server	192.168.43.141	192.168.43.141	
	Port	8888	9999	



::Uji Coba Kebenaran(2)::

Uji Coba Tombol *Keyboard* pada Android

No	Tombol Keyboard yang Ditekan	Hasil yang Keluar pada Komputer	Hasil
1	abcdefghijklmnopqrstuvwxyz	Mengetik huruf abcdefghijklmnopqrstuvwxyz	Sukses
2	1234567890	Mengetik angka 1234567890	Sukses
3	[Enter]	Pada layar teks kursor turun 1 baris	Sukses
4	[Tab]	Memindahkan kursor dalam 1 tabulasi ke arah kanan	Sukses
5	[Space]	Membuat jarak 1 karakter antar kata	Sukses
6	[Backspace]	Menghapus 1 karakter terakhir yang telah diketik	Sukses
7	[Delete]	File yang dipilih dihapus	Sukses
8	[Caps lock]	Sesuai dengan kondisi Caps lock	Sukses
9	[Ctrl] + a	Menjalankan perintah "Select all"	Sukses
10	[Ctrl] + c	Menjalankan perintah "Copy"	Sukses
11	[Ctrl] + v	Menjalankan perintah "Paste"	Sukses
12	[Ctrl] + x	Menjalankan perintah "Cut"	Sukses
13	[Ctrl] + z	Menjalankan perintah "Undo"	Sukses



::Uji Coba Kebenaran(3)::

Uji Coba Tombol dan Menggerakan *Mouse* pada Android

No	Gerakan pada Mouse	Hasil yang Keluar pada Komputer	Hasil
1	Menggerakan touchpad ke atas	Pointer bergerak ke atas	Sukses
2	Menggerakan touchpad ke bawah	Pointer bergerak ke bawah	Sukses
3	Menggerakan touchpad ke kanan	Pointer bergerak ke kanan	Sukses
4	Menggerakan touchpad ke kiri	Pointer bergerak ke kiri	Sukses
5	Menekan tombol L pada layar aplikasi	Klik kiri pada pointer	Sukses
6	Menekan tombol R pada layar aplikasi	Klik kanan pada pointer	Sukses
7	Menekan tombol V pada layar aplikasi	Scroll ke bawah	Sukses
8	Menekan tombol ^ pada layar aplikasi	Scroll ke atas	Sukses



..:KESIMPULAN:..

- Otentikasi menggunakan *username*, *password*, dan *MAC address* pada aplikasi dapat diimplementasikan dengan benar.
- *Keyboard* dapat dijalankan dengan baik, tombol angka, huruf, dan lain-lain sebagainya dapat berjalan dengan semestinya.
- *Mouse* dapat dijalankan dengan baik, *touchpad* serta tombol mouse seperti klik kiri, klik kanan, dan *scroll* dapat berjalan dengan benar.



..:SARAN:..

- Untuk pengembangan aplikasi lebih lanjut disarankan untuk dapat dikembangkan di dalam *platform* Windows Phone, IOS, Linux, dan Mac OSX.
- Untuk pengembangan aplikasi selanjutnya disarankan menggunakan *multi-touch* lebih dari dua jari sehingga fungsi-fungsi tombol khusus dapat diimplementasikan seperti ctrl+alt+delete dan lain-lain.
- Penambahan tombol-tombol khusus seperti fn, windows, F1, F2, F3 dan lain-lain.



The background of the slide is a repeating pattern of a logo. The logo consists of a light blue gear-like outer ring with a yellow lotus flower in the center. The text is centered over this pattern.

.:TERIMA KASIH:.

